Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

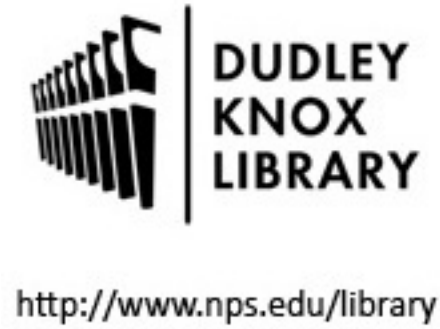
1986

Development of a data acquisition system to aid in the aerodynamics study of various helicopter configurations.

Witt, Patrick A.

http://hdl.handle.net/10945/21785

Downloaded from NPS Archive: Calhoun



http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF A DATA ACQUISITION SYSTEM TO AID IN THE AERODYNAMIC STUDY OF VARIOUS HELICOPTER CONFIGURATIONS

by

Patrick A. Witt

March 1986

Thesis Advisor: Donald M. Layton

Approved for public release; distribution is unlimited.

T227315

RITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release;
DECLASSIFICATION / DOWNGRADING SCHEDULE	Distribution is unlimited
ERFORMING ORGANIZATION REPORT NUMBER(S)	S. MONITORING ORGANIZATION REPORT NUMBER(S)
NAME OF PERFORMING ORGANIZATION val Postgraduate School 6b. OFFICE SYMBOL (If applicable) 67	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
ADDRESS (City, State, and ZIP Code) nterey, California 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

TITLE (Include Security Classification)
EVELOPMENT OF A DATA ACQUISITION SYSTEM TO AID IN THE AERODYNAMIC STUDY OF ARIOUS HELICOPTER CONFIGURATIONS

PERSONAL AUTHOR(S)
Witt, Patrick A.

TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
ngineer's Thesis	FROM _____ TO _____	1986 March	115

SUPPLEMENTARY NOTATION

COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Data Acquisition, Data Reduction, Data Analysis, Balance Calibration, Balance Modification, Landing Gear design, Interactive programs

ABSTRACT (Continue on reverse if necessary and identify by block number)
his thesis developed a data acquisition system to be used in conjunction ith the 3.5' x 5.0' low speed wind tunnel at the Naval Postgraduate School. nteractive graphic programs were developed to aid in data acquisition and nalysis. In addition, the internal balance that was designed by Major Scott air and Major Chris Sargent was redesigned to correct some problems encountered with the drag component. The balance was also instrumented to record he pitch and yaw moment components. A calibration rig was designed and constructed in order to evaluate the interactions of the different components. he equipment used and programs developed for data acquisition and analysis ere adequate. However, balance calibration revealed problems with the calibration rig and location of the roll component strain gage. Both of these roblems will have to be corrected before accurate readings can be expected rom this balance design.

0. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
a. NAME OF RESPONSIBLE INDIVIDUAL Prof. Donald M. Layton	22b. TELEPHONE (Include Area Code) 646-2997 22c. OFFICE SYMBOL 67-LN

D FORM 1473, 84 MAR 83 APR edition may be used until exhausted.
All other editions are obsolete.

Approved for public release; distribution is unlimited.

Development of a Data Acquisition System to aid in the Aerodynamic Study of Various Helicopter Configurations

by

Patrick A. Witt
Lieutenant, United States Navy
B.S., United States Naval Academy, 1978

Submitted in partial fufillment of the
requirements for the degrees of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING
and
AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
March 1986

ABSTRACT

This thesis developed a data acquisition system to be used in conjunction with the 3.5' x 5.0' low speed wind tunnel at the Naval Postgraduate School. Interactive graphic programs were developed to aid in data acquisiton and analysis. In addition, the internal balance that was designed by Major Scott Mair and Major Chris Sargent was redesigned to correct some problems encountered with the drag component. The balance was also instrumented to record the pitch and yaw moment components. A calibration rig was designed and constructed in order to evaluate the interactions of the different components. The equipment used and programs developed for data acquisition and analysis were adequate. However, balance calibration revealed problems with the calibration rig and location of the roll component strain gage. Both of these problems will have to be corrected before accurate readings can be expected from this balance design.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. BACKGROUND

This project used the 3.5' x 5.0' low speed wind tunnel at the Naval Postgraduate School to continue the aerodynamic study of the effects of helicopter noses and tails on drag conducted by Major Mair [Ref. 1] and Major Sargent [Ref. 2]. Their work included the construction of nine various helicopter configurations, a sting mounted support system and a four-degree-of-freedom balance that was internal to the model. They also studied the airflow around the various configurations with cotton tufting.

To reduce and analyze the data, Majors Mair and Sargent developed several independent computer programs. These programs calibrated the balance, corrected the raw data from the wind tunnel runs, computed the equivalent flat plate area and produced plots of various parameters for comparison. However, due to problems encountered with the drag component of the balance, their results were inconclusive.

B. GOALS

The primary goal of this project was to provide the students enrolled in the helicopter design class at the Naval Postgraduate School with a laboratory type set-up that would develop realistic Equivalent Flat Plate Area

information for various helicopter configurations. This included creating interactive computer programs that the students could use with the personal computer located at the wind tunnel.

The secondary goals of this project were to upgrade the two-axis internal wind tunnel balance to a six-axis internal balance and to develop a calibration rig to evaluate the interactions of the balance components.

Three landing gear configurations were also designed and constructed for future analysis of the drag that they add to the helicopter.

II. APPROACH TO THE PROBLEM

A. LANDING GEAR DESIGN

To provide a realistic representation of landing gear used with modern-day helicopters, one type of landing gear was selected for each nose shape (Figures 2.1-2.3). Planviews for the three types of landing gear were prepared and are included in Appendix A.

For the attack nose, a skid type of landing gear was constructed of aluminum tubing secured to an aluminum plate. This type of landing gear is considered a fixed gear but was chosen because of its wide use for numerous helicopters. For both the smooth nose and blunt nose a simulated retractable gear was constructed of wood stubwings and model airplane tires. Threaded inserts were mounted in the noses and stubwings to allow easy removal of the wheel and strut assemblies. This allowed the models to be tested in both the clean and dirty configuration.

B. INTERNAL BALANCE MODIFICATION

A modified Mair/Sargent balance, Figures 2.4-2.7, was used for this project.

To improve the output recorded from the axial component, the cuts (see Figure 2.4) alongside that cavity were increased by 1/16 of an inch. In addition, the cavity itself was squared off thus reducing the curvature of the

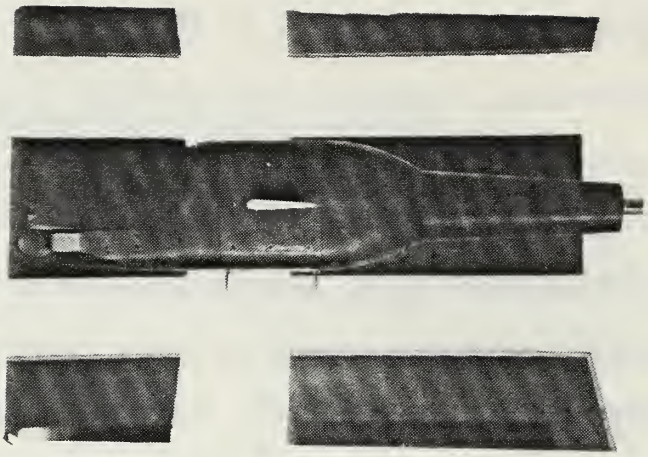


Figure 2.1 Attack Nose with gear

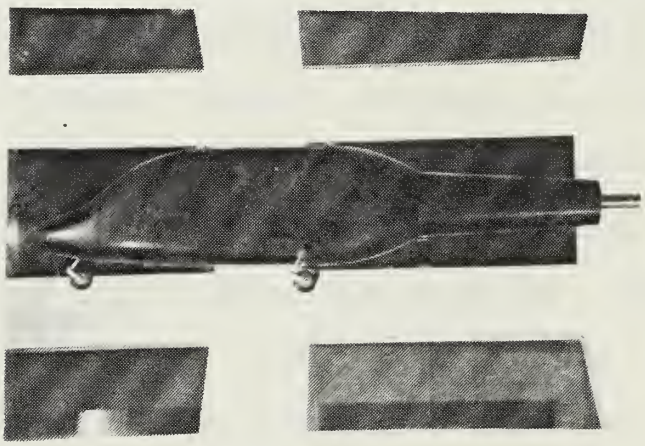


Figure 2.2 Smooth Nose with gear

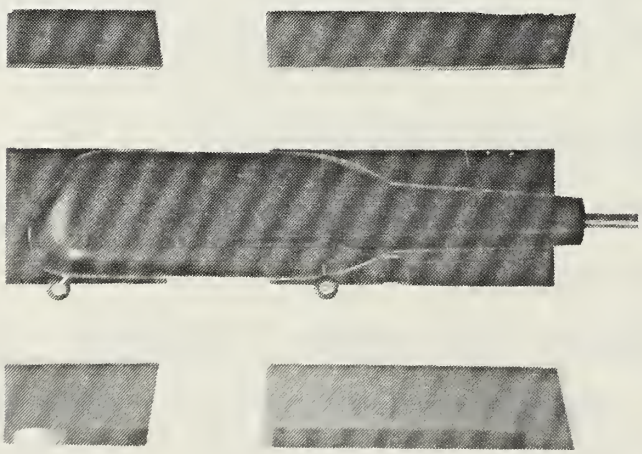


Figure 2.3 Blunt Nose with gear

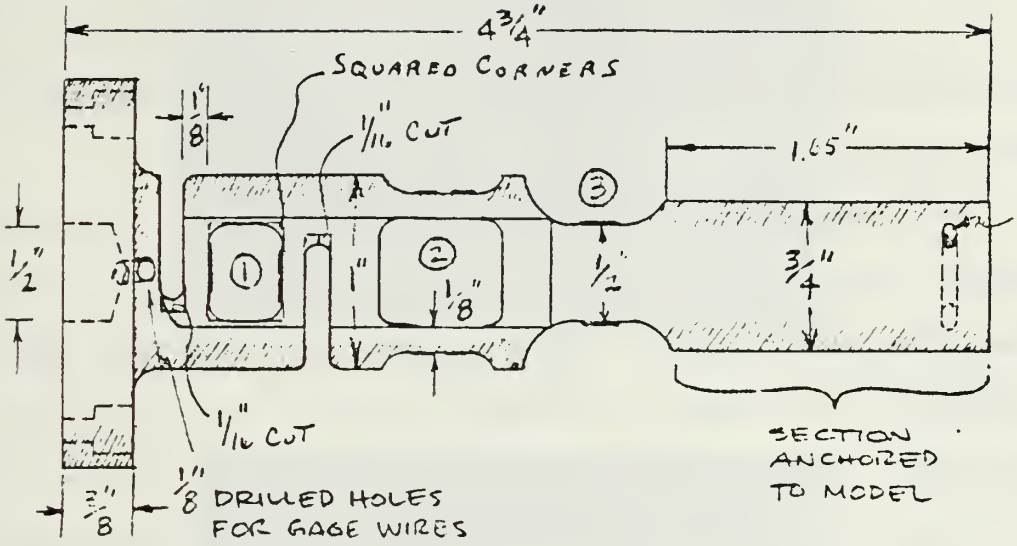


Figure 2.4 Internal Balance with Modifications

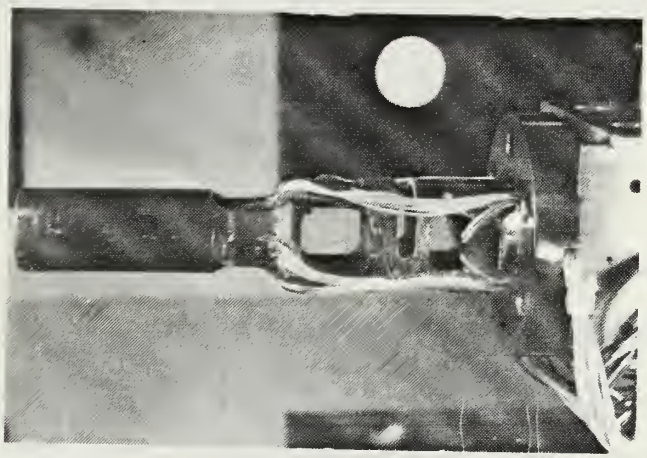


Figure 2.5 Re-Wired Internal Balance

surfaces in the cavity. These modifications increased the flat area upon which the strain gages could be placed.

An area (see Figure 2.6), similar to that for the pitching moment component, was cut to allow recording of the yawing moment component.

For compatability and increased accuracy, the aluminum strain gages were removed and replaced by EA-09-062AQ-350 stainless steel ones. These gages were smaller and thus allowed for better placement within the cavities. To provide a longer life, the gages were bonded to the balance with an M-Bond AE-15 adhesive system. They were cured at a temperature of 150 degrees Fahrenheit for two hours.

It was desired to record the six component forces on the helicopter; lift, drag, yaw, pitching moment, yawing moment and rolling moment. However, since the internal balance was designed to record only four components, the sting support was instrumented to record the yaw force and rolling moment (Figure 2.8).

The gages for the yaw component were placed on the side of the sting support to undergo tension and compression when subjected to a yawing force. The gages for the rolling moment component were placed on top and bottom of the sting support at a 45 degree angle to the sting axis. Thus, they experienced tension and compression when the model was subjected to a rolling moment.

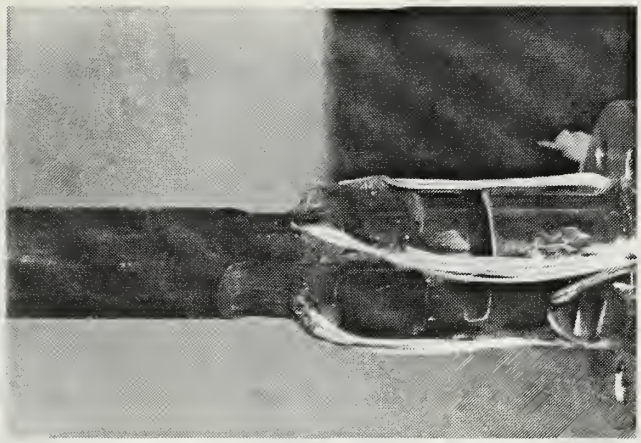


Figure 2.6 Lift, Pitch Moment and Yaw Moment Gages

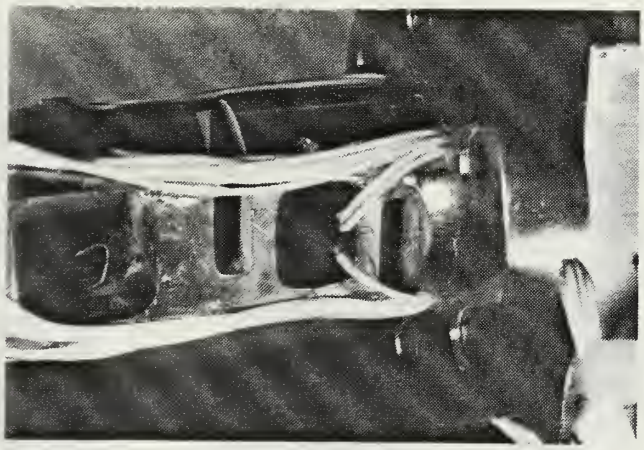


Figure 2.7 Drag Component Gage

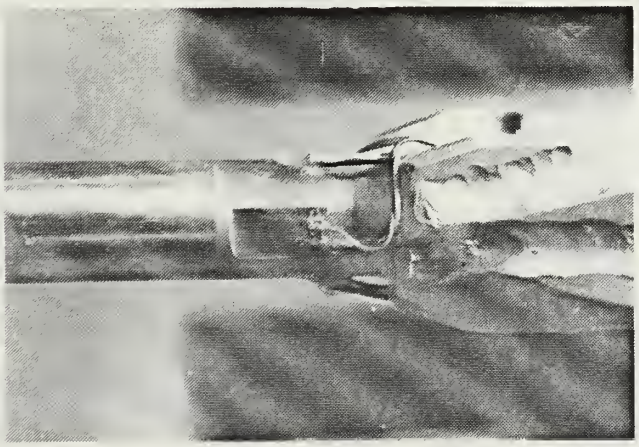


Figure 2.8 Yaw and Roll Moment Gages

C. BALANCE CALIBRATION

In order to determine any interaction between the six recording components, a calibration rig was designed and mounted to the main fuselage (Figure 2.9). The pans and pullies were arranged to simulate the twelve forces and moments that the helicopter models would experience in the wind tunnel. Each pan, or component was loaded from zero to twenty pounds in one pound increments. With each loading, readings from the other five components were recorded. This method was conducted until all six components were loaded in both the positive and negative direction and produced thirty interaction matrices.

A balance calibration program, Figure A.16, was written to determine the relations required to convert the raw data counts to actual forces and moments. For each component loading, the prime gage constants were determined using the following least squares curve fit [Ref. 3]:

$$[\text{Sum}(Xi)^2]*K1 + [\text{Sum}(Xi^3)]*K2 = \text{Sum}(Xi*Yi)$$

$$[\text{Sum}(Xi)^3]*K1 + [\text{Sum}(Xi^4)]*K2 = \text{Sum}((Xi^2)*Yi)$$

where Xi is the raw data count
and Yi is the applied load

Once these constants were computed for all twelve loadings, the raw data counts were converted to forces and moments. Then for each interaction matrix, the same least squares equations were used to determine the interaction coefficients. These prime gage constants and interaction

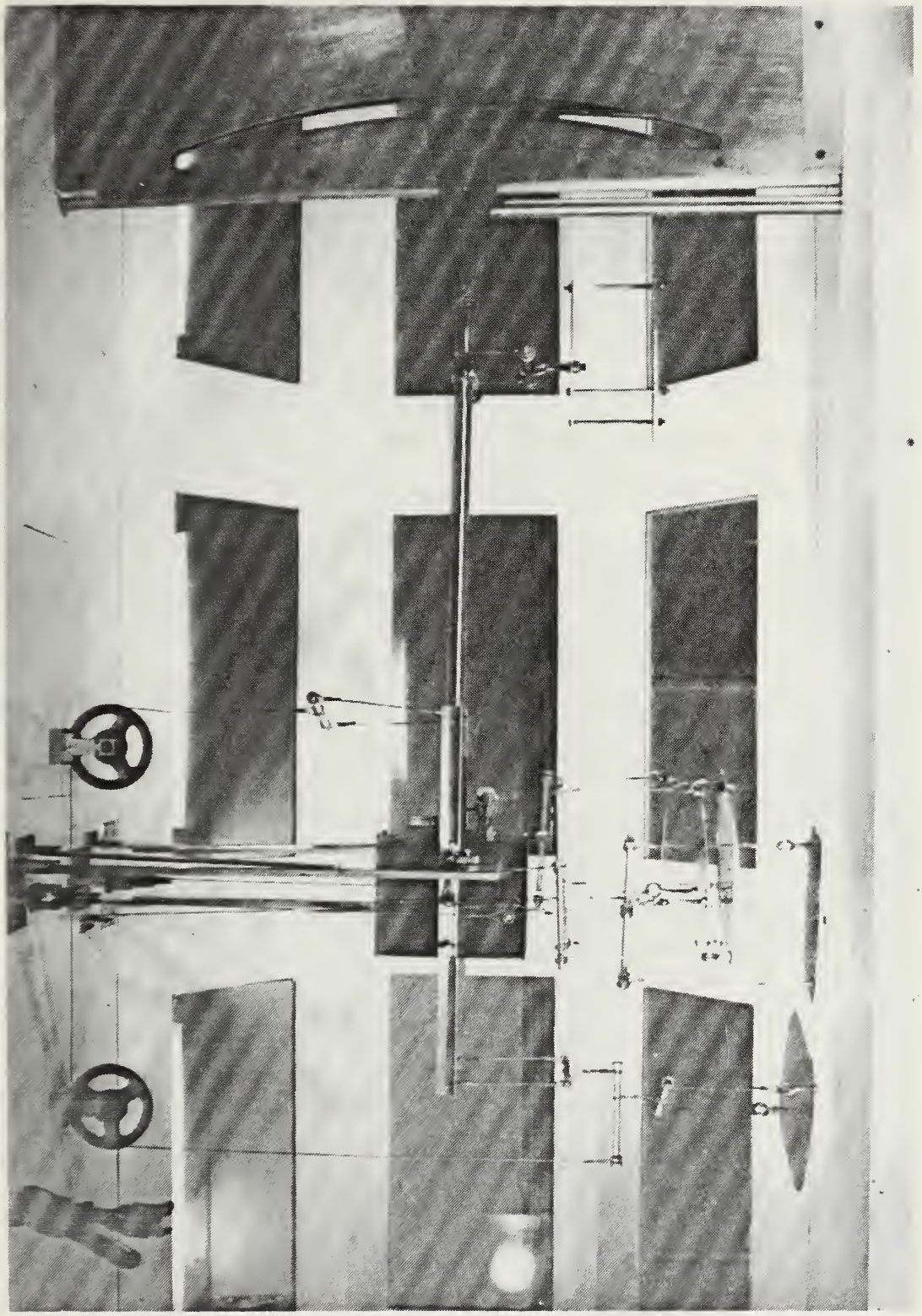


Figure 2.9 Balance Calibration Rig

coefficients were stored for use in the reduction of the raw data recorded during the wind tunnel tests.

D. DATA ACQUISITION

To improve data reduction, it was desired to have a data acquisition system built into the wind tunnel instrumentation. The strain gages on the internal balance and support sting were connected to a Pacific Instruments model 8255 transducer amplifier (Figure 2.10). The outputs from this unit were then routed to data acquisition cards mounted in an IBM PC AT. A data acquisition program, Figure A.10, was incorporated into the interactive program set that allowed for quick and easy recording of test data.

Having the transducer amplifier connected to the strain gages allowed for easy zeroing and balancing of the gages. The data acquisition set-up provided real time readout of the forces that the helicopter was experiencing in the wind tunnel.

Figure 2.10 Data Acquisition Equipment

III. SOLUTION TO THE PROBLEM

A. DATA COLLECTION AND REDUCTION

The data collection program, Figure A.10, was designed to record the data in a standardized collection method. Interactive steps instruct the user to follow the procedures listed below.

1). Zero all amplifiers without the model in place.
2). Mount the model to the sting support and record a zero force reading.
3). Place the calibration switches to the + position and record a calibration reading.
4). Replace the calibration switchres to the center position and start the wind tunnel.
5). Record first and last data point at zero angle of attack with all other readings between plus ten degrees and minus eight degrees angle of attack.

By using a delay loop, changes could be made to the angle of attack of the model before the next set of data points were recorded. For each tunnel speed, the data recorded for the various angles of attack were stored on separate files.

Once the tunnel runs were completed and the data was recorded on file, another program, Figure A.11, was designed to convert the raw readings to readings of forces and monemts. The sixty interaction coefficients developed from the balance calibration were incorporated into this data reduction. The following equations, obtained from Mr.

David Backs at the NASA AMES Reseasrch Center, were used to correct for the interaction of the balance components.

```
F1' = F1 - (delF1/delF2)*F2 - (delF1/delF3)*F3 - .........
     - (delF1/delF2^2)*F2^2 - ....... -(delF1/delF6^2)*F6^2

F1''= F1 - (delF1/delF2)*F2'-.............................
     - (delF1/delF2^2)*F2'^2 ....... -(delF1/delF6^2)*F6'^2
```

These equation were written for each component and placed in an iterative loop that checked the difference between the two prime values. After the forces were corrected for component interaction, a weight tare equation was used to correct for the weight of the model

B. ANALYSIS OF DATA

The main file of the interactive programs, Figure A.9, controlled the data recording, data reduction and data analysis with a menu format. For data analysis, the stored converted data readings were used to calculate the lift and drag coefficients and equivalent flat plate area for each angle of attack. Files were created for coefficient of lift versus angle of attack, coefficient of drag versus coefficient of lift, coefficient of drag versus coefficient of lift squared and equivalent flat plate area versus angle of attack. There was one file of each created for each tunnel speed.

A plotting routine, Figure A.15, was included in the interactive programs to allow quick analysis of the recorded data. To aid in the analysis, up to three plots

could be shown on one graph. A delay loop was incorporated into the plotting routine to allow for the option of obtaining a hard copy of the graph by using the print screen command.

IV. RESULTS

A. BALANCE CALIBRATION

The balance was loaded in the twelve component directions and the balance calibration program was used to produce the tables B.2 through B.13. During the balance calibration, a drift in each of the components was noted under steady state conditions. To correct for this problem it was assumed that each component had a constant drift rate. The difference between the first and last zero reading was divided by the number of data points taken. This correction factor was then applied to each data point. This method produced good correlation between different data points taken for the same load.

B. DATA COLLECTION

Test runs were conducted to evaluate the data acquisition program. The vibrations of the model in the wind tunnel resulted in erratic fluctuations in the sense indicators of the amplifiers. To correct for this problem, the acquisition program was modified to collect one hundred samples at a rate of five hundred samples per second. The average of these one hundred samples was taken as one data point. This method produced constant readings for data points of similar conditions.

C. DATA REDUCTION AND ANALSYIS

The data reduction program was used to convert the raw test data. The interaction equations diverged instead of converging to a single value. This pointed out a problem with the interaction coefficients. Examination of the calibration tables revealed extremely large interactions between the loading of the lift component and reaction in the roll moment component. There was also noted a large interaction between the loading of the pitch moment component and reaction in the lift component.

The reduction program was modified to correct for the interactions between the lift and drag components only. The reduction program was again executed using the recorded test data. This time the interactions converged. This proved the validity of the interaction equations and confirmed the problems with the balance calibration.

A test file was created to display the plotting capabilities of the interactive programs. Figure 4.1 shows the results.

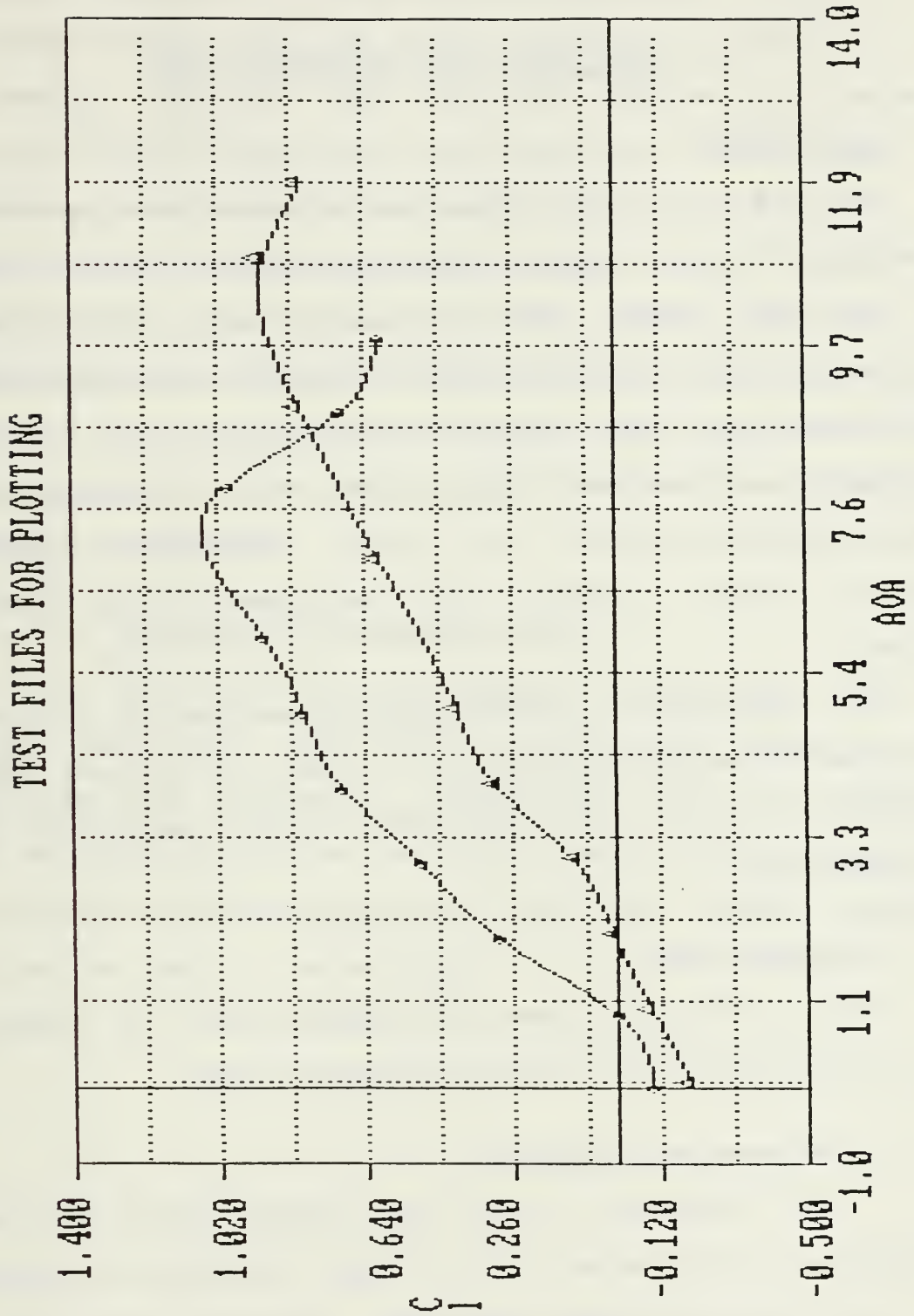


Figure 4.1 Output from Plotting Routine

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The data acquisition system and programs developed were adequate for recording the forces and moments experienced in the wind tunnel. The extremely large interactions between the lift and roll moment components and between the pitch moment and lift components (see Tables B.2 - B.13) precluded any analysis of wind tunnel data. However, the data acquisition system and balance calibration program warrant further research and development of the balance system.

The interactive programs greatly reduced the workload required in the data acquisition and analysis phases. They provided a quick and easy means for the analysis of the recorded data.

B. RECOMMENDATIONS

The following are given as recommendations to improve the calibration of the internal balance system.

1. Balance Modification

The location of the roll moment strain gage was decided to be the cause of the large interaction between the lift and roll moment component. This gage was located on the back end of the sting support system. The roll

moment component was calibrated by applying a torque to the center section of the model that was mounted to the internal balance. When the lift component was loaded, a large bending moment was felt by the roll moment gage and thus producing the large interactions.

By placing the roll moment strain gage at a forty-five degree angle on the same cut-out section as the pitch moment gage, the torque applied to the center section can be used for calibration. Also, the loading of the lift component will not greatly affect the reaction of the roll moment component.

2. Calibration Rig Modification

By using the center section of the model to mount the calibration pans for the pitch moment component, the strains were incorrectly transmitted to the lift component. A separate calibration set-up will have to be designed that isolates the force appplied to the pitch moment component from the rest of the balance.

APPENDIX A

SKETCHES AND PROGRAMS

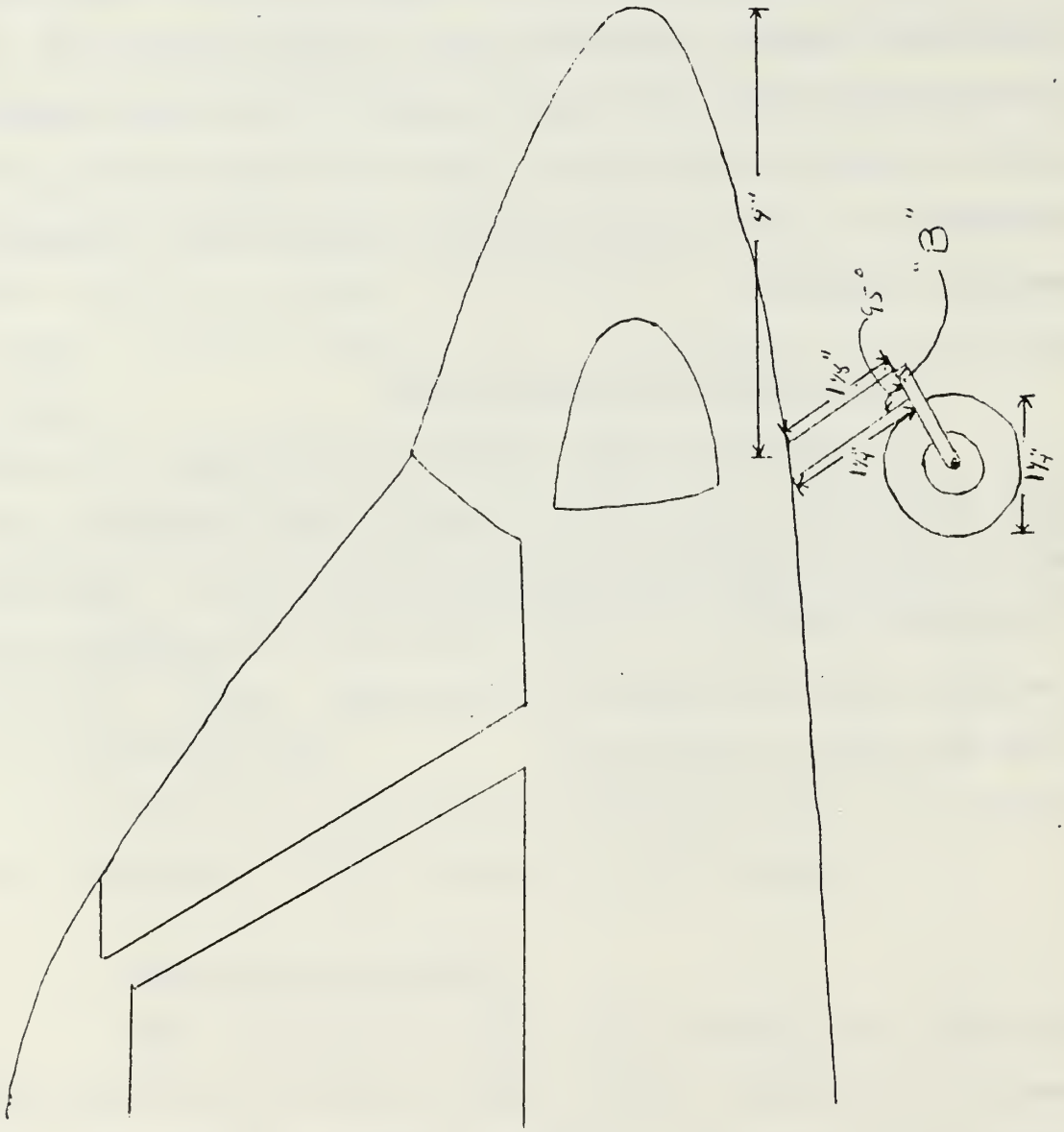


Figure A.1 Nose Gear for Smooth Nose

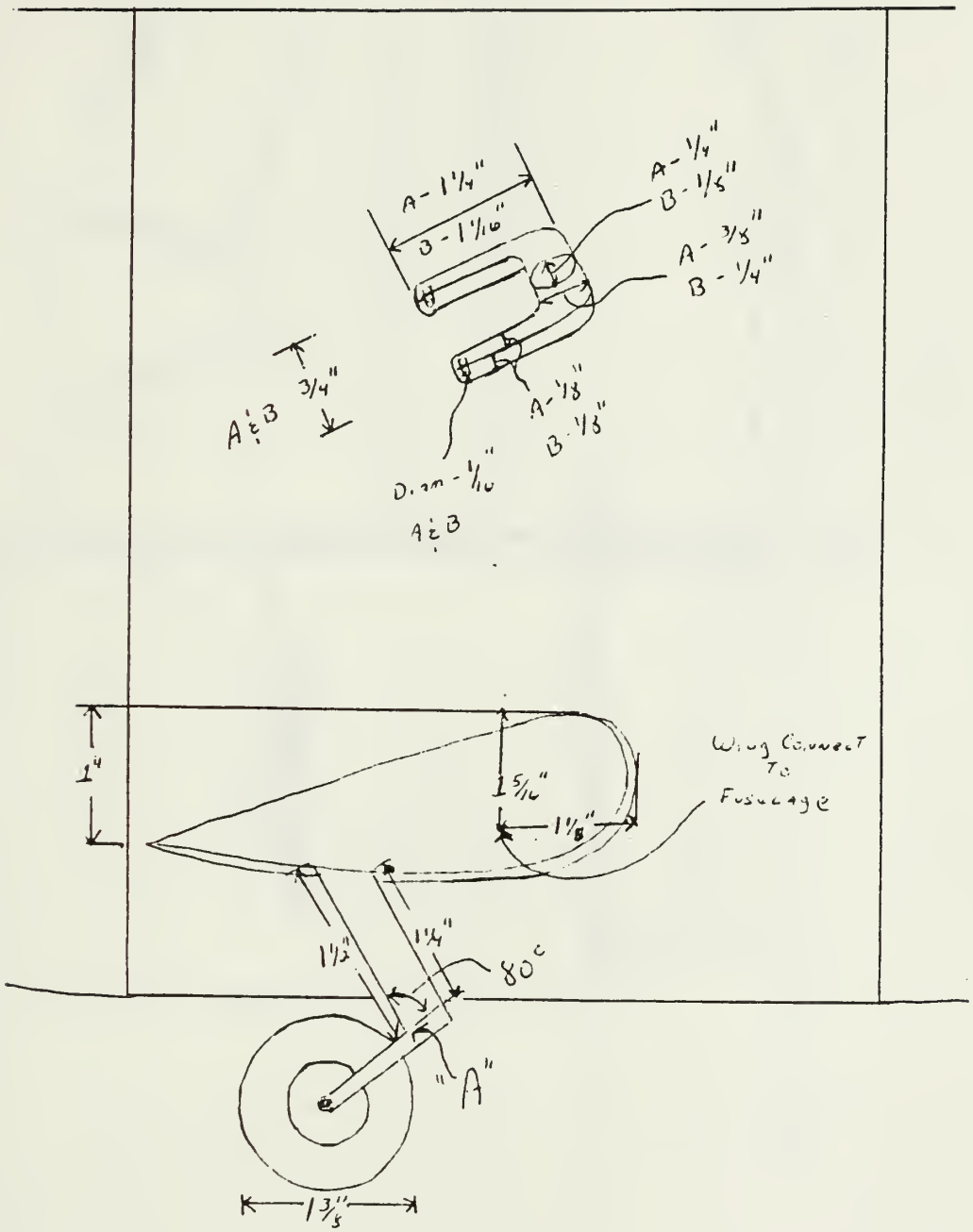


Figure A.2 Main Gear for Smooth Nose

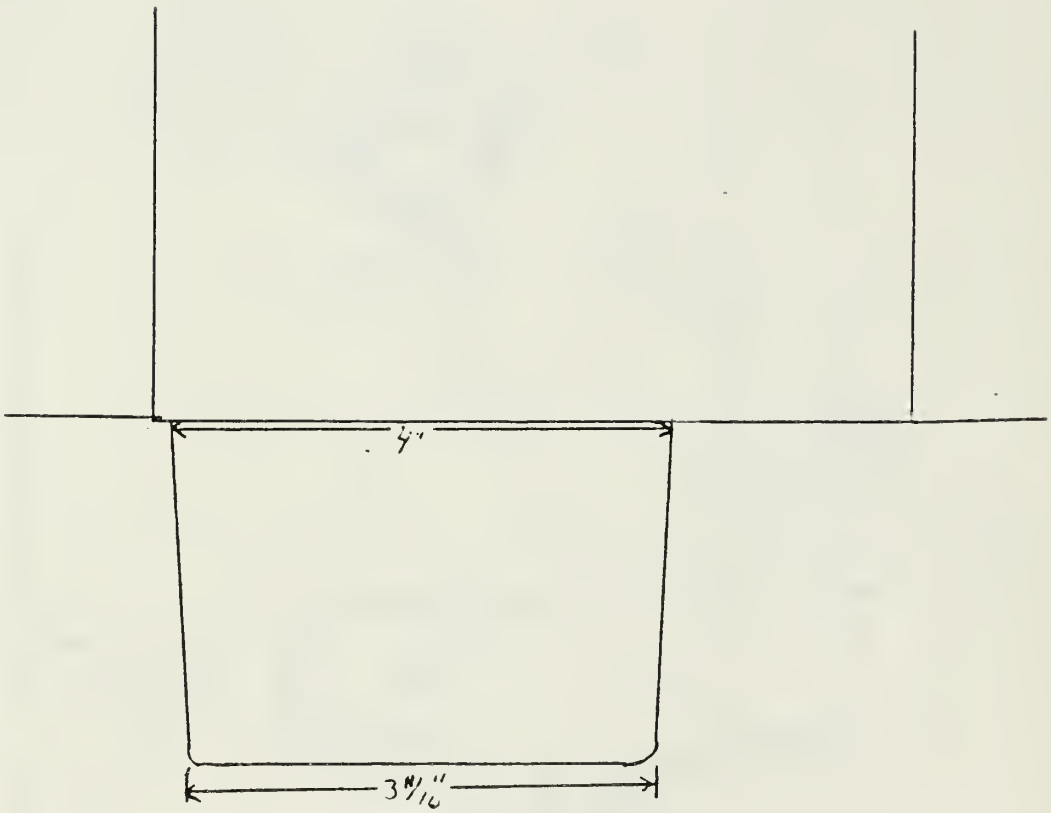


Figure A.3 Stubwing for Smooth Nose (Top View)

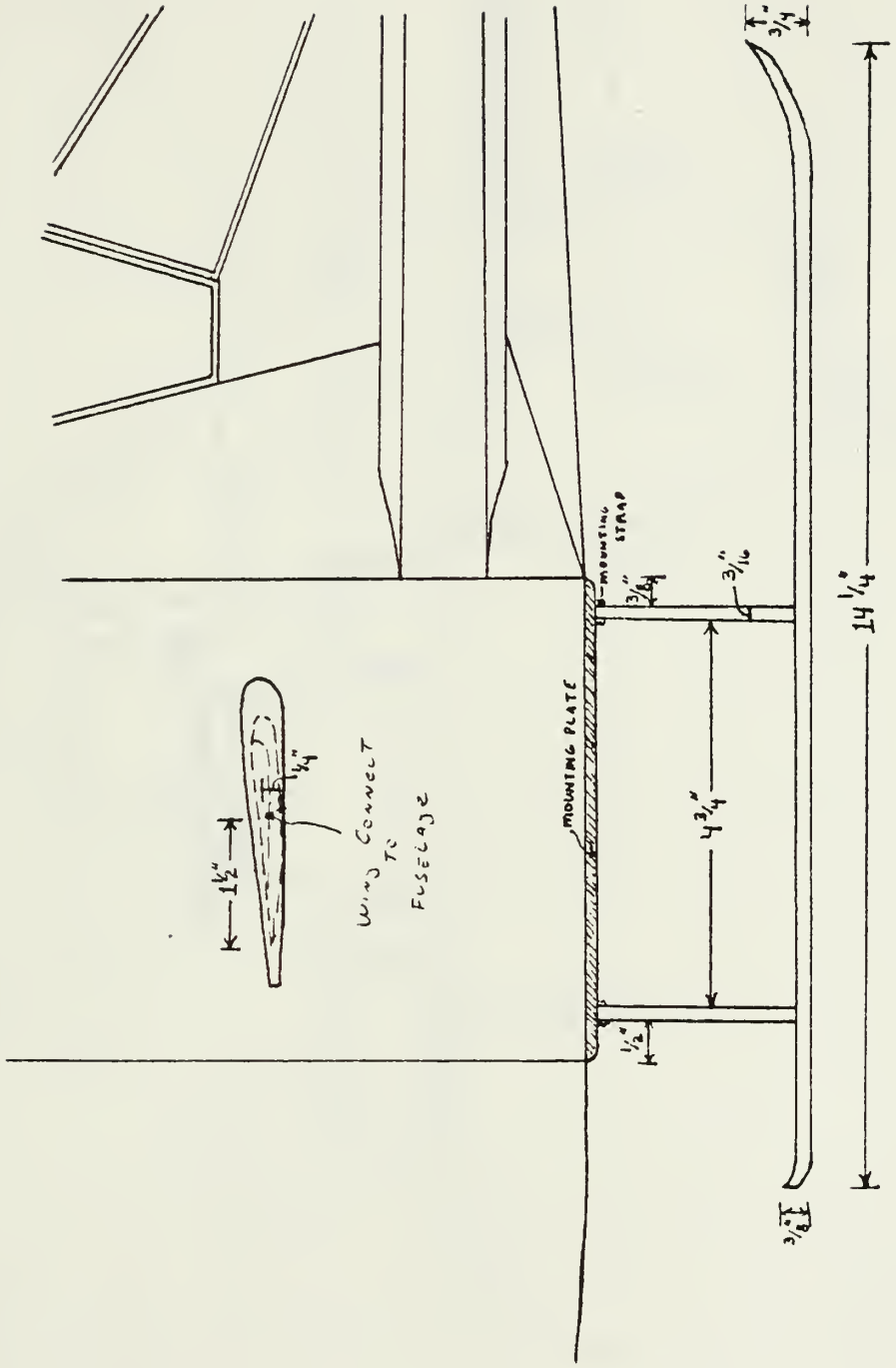


Figure A.4 Skid Gear and Wing for Attack Nose

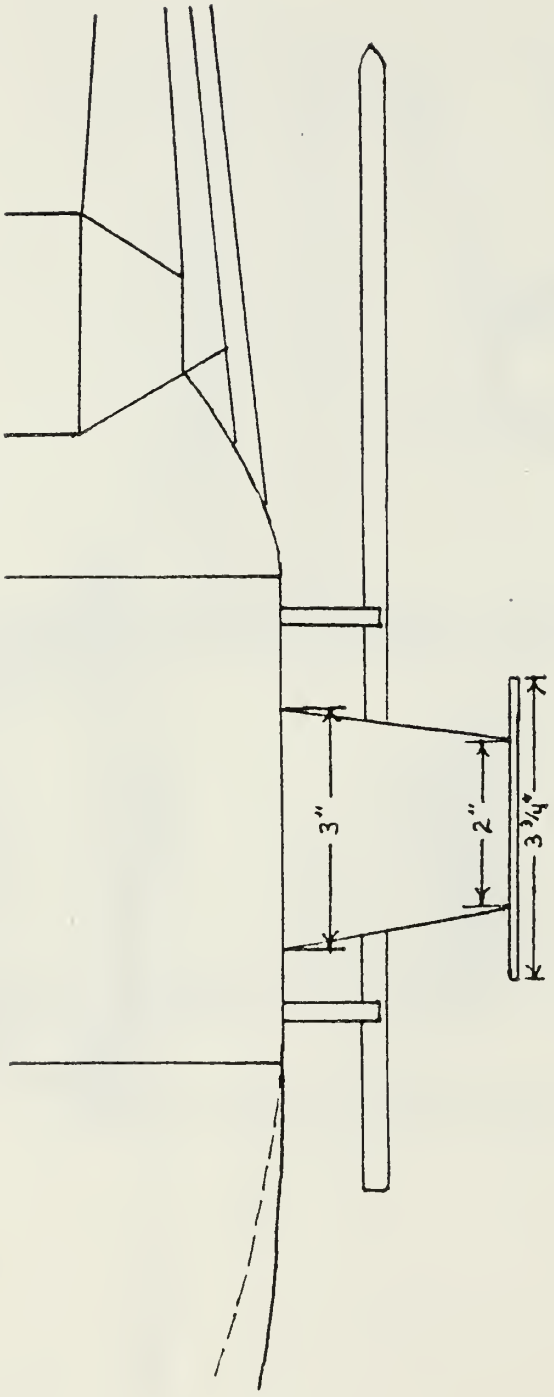


Figure A.5 Skid Gear for Attack Nose (Top View)

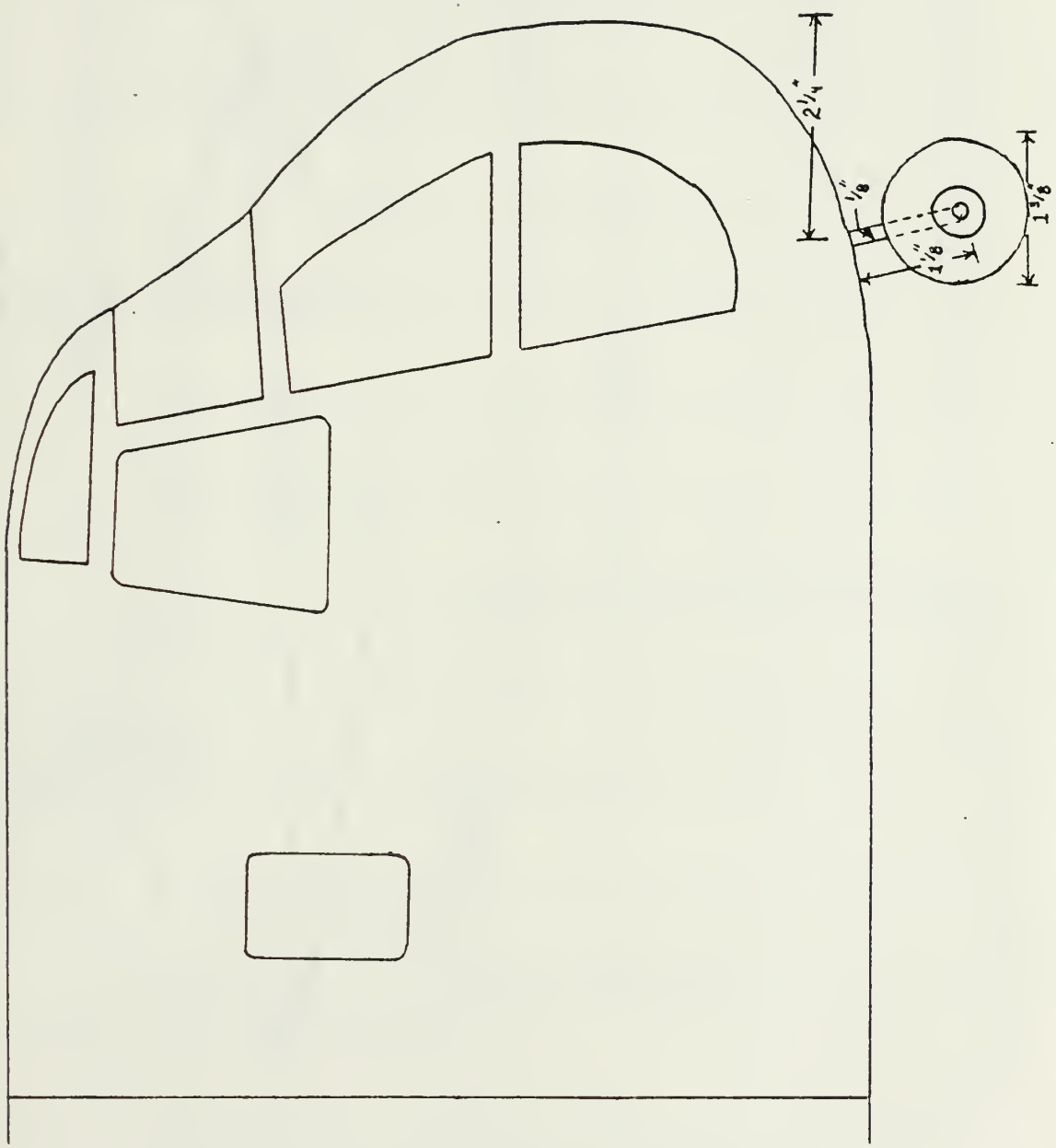


Figure A.6 Nose Gear for Blunt Nose

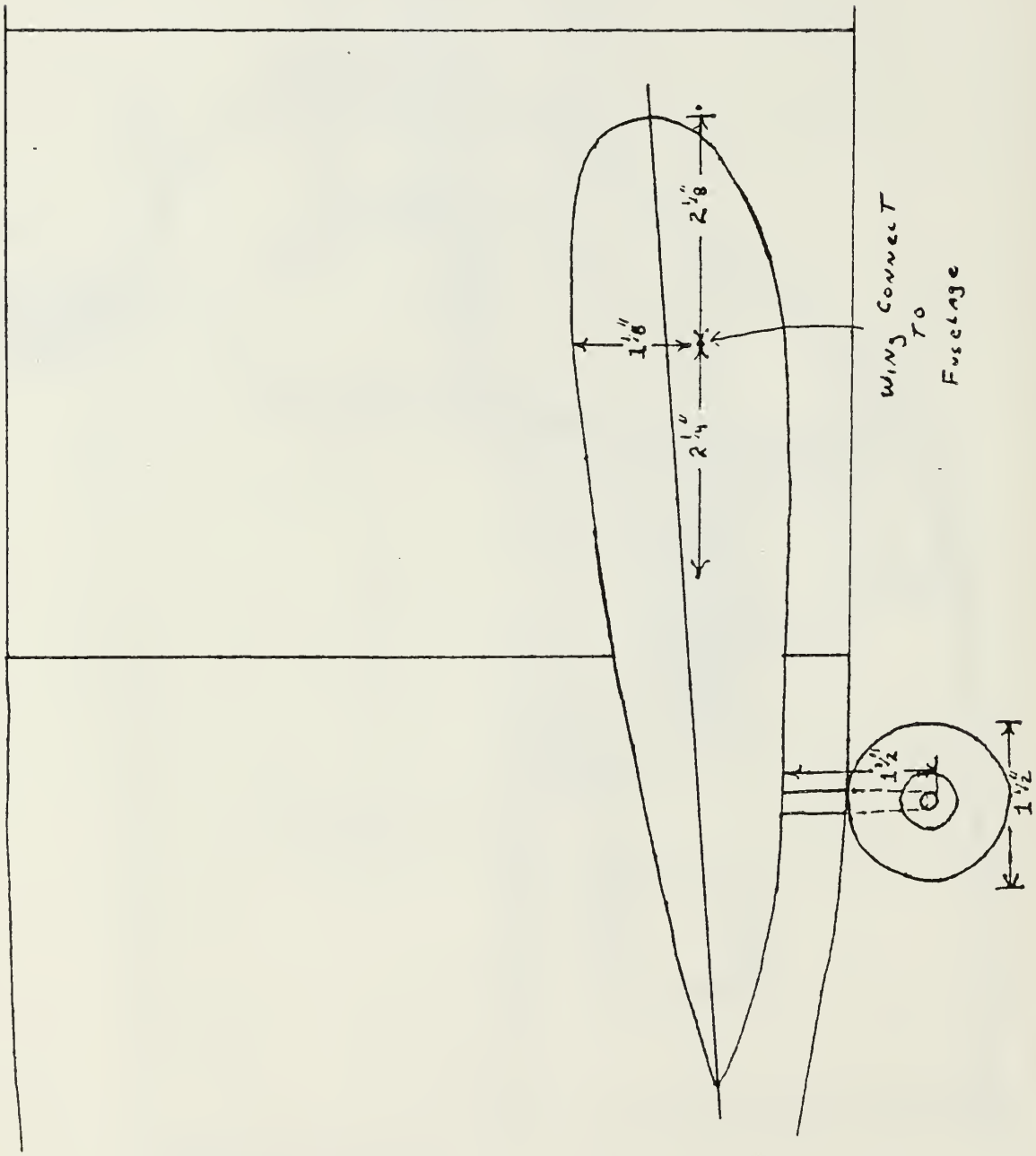


Figure A.7 Main Gear for Blunt Nose

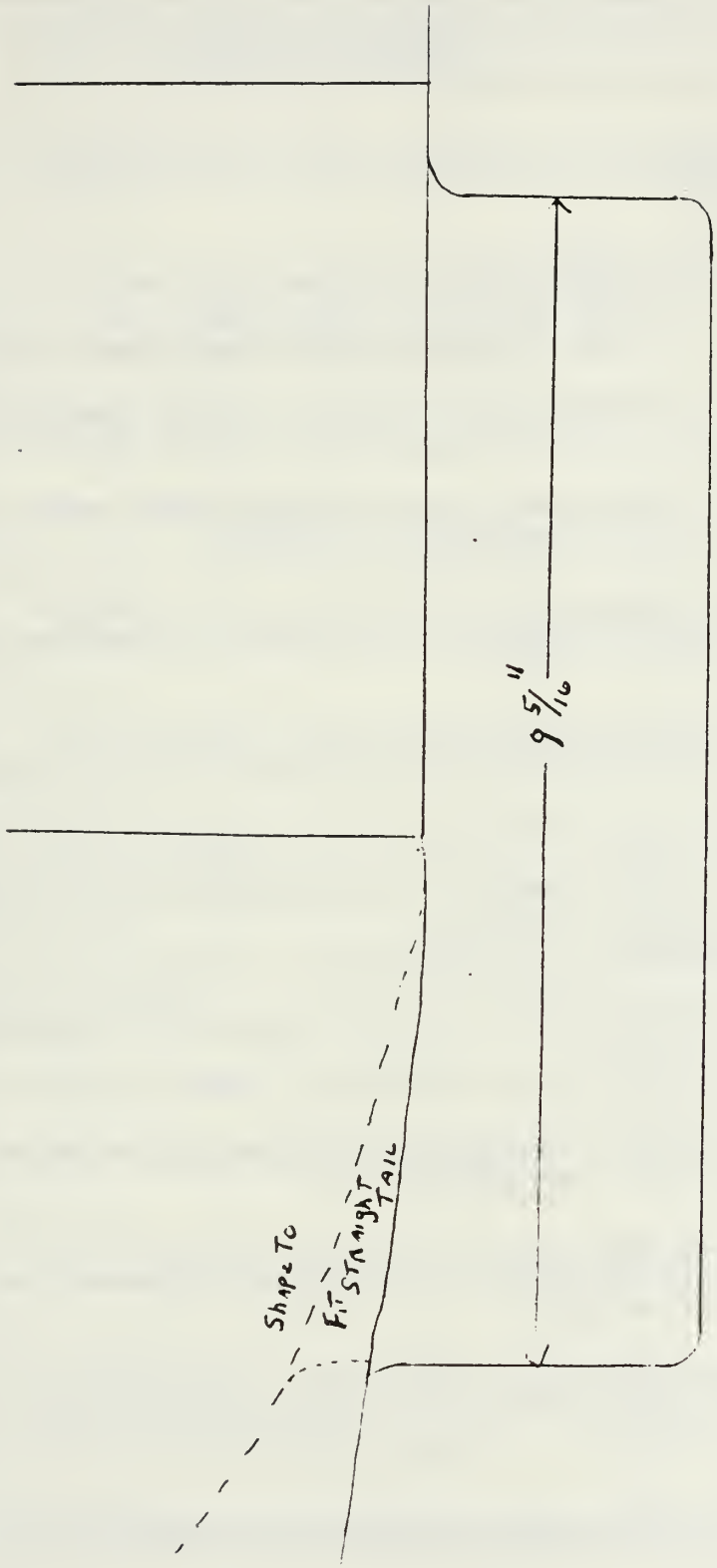


Figure A.8 Stubwing for Blunt Nose (Top View)

```
10   REM------MAIN.BAS (MAIN CONTROL PROGRAM)-----
20   '                  PATRICK A. WITT
30   '                    20 JULY 1985
40   '-----------------------------------------
50   REM
280  KEY OFF
290  SCREEN 1
300  COLOR 1
310  FOR I = 1 TO 2
320  IF I = 1 THEN FILL = 11 ELSE FILL = 12
330  IF I = 1 THEN EDGE = 9 ELSE EDGE = 10
340  IF I = 1 THEN X1=106: X2=0: X3=56 ELSE X1=108: X2=2:
     X3=58
350  IF I = 1 THEN Y1=48: Y2=94: Y3=140 ELSE Y1=46: Y2=92:
     Y3=138
360  ' "H"
370  DRAW "C=EDGE;BM=X1;,=Y1;U30R4D12R12U12R4D30L4U12"
380  DRAW "L12D12L4BE1P=FILL;,=EDGE;"
390  ' "E"
400  X1= X1 + 26
410  DRAW "C=EDGE;BM=X1;,=Y1;U30R20D4L16D9R8D4L8D9R16D4L2
     0BE1P=FILL;,=EDGE;"
420  ' "L"
430  X1= X1 + 26
440  DRAW "C=EDGE;BM=X1;,=Y1;U30R4D26R16D4L20BE1P=FILL;,
     =EDGE;"
450  ' "O"
460  X1= X1 + 26
470  DRAW "C=EDGE;BM=X1;,=Y1;U30R20D30L20BR4BU4U22R12D22L
     12BG1P=FILL;,=EDGE;"
480  'NEXT LINE
490  ' "C"
500  DRAW "C=EDGE;BM=X2;,=Y2;U30R20D4L16D22R16D4L20BE1P=
     FILL;,=EDGE;"
510  ' "O"
520  X2= X2 + 26
530  DRAW "C=EDGE;BM=X2;,=Y2;U30R20D30L20BR4BU4U22R12D22L
     12BG1P=FILL;,=EDGE;"
540  ' "N"
550  X2= X2 + 26
560  DRAW "C=EDGE;BM=X2;,=Y2;U30R4F12U12R4D30L4H12D12L4BE
     1P=FILL;,=EDGE;"
570  ' "F"
580  X2= X2 + 26
590  DRAW "C=EDGE;BM=X2;,=Y2;U30R20D4L16D6R8D4L8D16L4BE1P=
     FILL;,=EDGE;"
600  ' "I"
610  X2= X2 + 26
620  DRAW "C=EDGE;BM=X2;,=Y2;U30R4D30L4BE1P=FILL;,=EDGE;"
630  ' "G"
```

Figure A. 9 MAIN.BAS - Main Controlling Program

```
640  X2= X2 + 14
650  DRAW "C=EDGE;BM=X2;,=Y2;U30R20D4L16D22R12U4L4U4R8"
660  DRAW "D12L20BE1P=FILL;,=EDGE;"
670  ' "U"
680  X2= X2 + 26
690  DRAW "C=EDGE;BM=X2;,=Y2;U30R4D26R12U26R4D30L20BE1P=
     FILL;,=EDGE;"
700  ' "R"
710  X2= X2 + 26
720  DRAW "C=EDGE;BM=X2;,=Y2;U30R20D15L12F15L4H15D15L4BU1
     9BR4U7"
730  DRAW "R12D7L12BL1P=FILL;,=EDGE;"
740  ' "A"
750  X2= X2 + 30
760  DRAW "C=EDGE;BM=X2;,=Y2;U30R20D30L4U15L12D15L4BU19BR
     4U7"
770  DRAW "R12D7L12BL1P=FILL;,=EDGE;"
780  ' "T"
790  X2= X2 + 26
800  DRAW "C=EDGE;BM=X2;,=Y2;BU26U4R20D4L8D26L4U26L8BE1P=
     FILL;,=EDGE;"
810  ' "I"
820  X2= X2 + 26
830  DRAW "C=EDGE;BM=X2;,=Y2;U30R4D30L4BE1P=FILL;,=EDGE;"
840  ' "O"
850  X2= X2 + 14
860  DRAW "C=EDGE;BM=X2;,=Y2;U30R20D30L20BR4BU4U22R12D22L
     12BG1P=FILL;,=EDGE;"
870  ' "N"
880  X2= X2 + 26
890  DRAW "C=EDGE;BM=X2;,=Y2;U30R4F12U12R4D30L4H12D12L4BE
     1P=FILL;,=EDGE;"
900  'NEW LINE
910  ' "A"
920  DRAW "C=EDGE;BM=X3;,=Y3;U30R20D30L4U15L12D15L4BU19BR
     4U7"
930  DRAW "R12D7L12BL1P=FILL;,=EDGE;"
940  ' "N"
950  X3= X3 + 26
960  DRAW "C=EDGE;BM=X3;,=Y3;U30R4F12U12R4D30L4H12D12L4BE
     1P=FILL;,=EDGE;"
970  ' "A"
980  X3= X3 + 26
990  DRAW "C=EDGE;BM=X3;,=Y3;U30R20D30L4U15L12D15L4BU19BR
     4U7"
1000 DRAW "R12D7L12BL1P=FILL;,=EDGE;"
1010 ' "L"
1020 X3= X3 + 26
1030 DRAW "C=EDGE;BM=X3;,=Y3;U30R4D26R16D4L20BE1P=FILL;,=
     EDGE;"
```

Figure A.9 MAIN.BAS (cont.)

```
1040 ' "Y"
1050 X3= X3 + 26
1060 DRAW "C=EDGE;BM=X3;,=Y3;BU30BL1R5F9R4E9R5G14D16L4"
1070 DRAW "U16H14BR2BD1P=FILL;,=EDGE;"
1080 ' "S"
1090 X3= X3 + 40
1100 DRAW "C=EDGE;BM=X3;,=Y3;BU1U2E2R16H19U4E2R20F2D2G2"
1110 DRAW "L16F19D4G2L20H2BR2BU1P=FILL;,=EDGE;"
1120 ' "I"
1130 X3= X3 + 32
1140 DRAW "C=EDGE;BM=X3;,=Y3;U30R4D30L4BE1P=FILL;,=EDGE;"
1150 ' "S"
1160 X3= X3 + 14
1170 DRAW "C=EDGE;BM=X3;,=Y3;BU1U2E2R16H19U4E2R20F2D2G2"
1180 DRAW "L16F19D4G2L20H2BR2BU1P=FILL;,=EDGE;"
1190 NEXT I
1200 LOCATE 22,4
1210 INPUT "WOULD YOU LIKE INSTRUCTIONS (Y/N)";Z$
1220 IF Z$ = "Y" OR Z$ = "y" THEN GOTO 1240
1230 GOTO 2000
1240 SCREEN 0
1250 WIDTH 80
1260 COLOR 15,1
1270 CLS
1280 PRINT
1290 PRINT TAB(15)"THIS  IS A MENU  DRIVEN  PROGRAM THAT
     WILL ALLOW YOU TO"
1300 PRINT TAB(10)"EVALUATE DIFFERENT  HELICOPTER CONFIGU
     RATIONS  AND DETERMINE"
1310 PRINT TAB(10)"THEIR LIFT  AND DRAG COEFFICIENTS AND
     THEIR EQUIVALENT FLAT"
1320 PRINT TAB(10)"PLATE AREA.  THE DATA GENERATED  DURING
     THE TUNNEL RUNS WILL"
1330 PRINT TAB(10)"BE USED WITH THIS PROGRAM."
1340 PRINT TAB(15)"THE  FIRST  OPTION  IS  USED  TO RECORD
     THE DATA DURING"
1350 PRINT TAB(10)"THE  TUNNEL RUNS.  THE SECOND  OPTION
     CONVERTS  THE RECORDED"
1360 PRINT TAB(10)"RAW COUNTS TO  FORCES AND MOMENTS.  YOU
     WILL HAVE THE CHANCE"
1370 PRINT TAB(10)"TO PROVIDE NAMES  FOR  EACH OF THESE
     FILES AS YOU GO ALONG."
1380 PRINT TAB(10)"THE THIRD OPTION  LETS YOU EXAMINE THE
     FORCES RECORDED.  IT"
1390 PRINT TAB(10)"WILL BE OF NO USE TO YOU TO EXAMINE THE
     RAW COUNTS."
1400 PRINT TAB(15)"THE FOURTH  OPTION USES  THE CONVERTED
     DATA TO COMPUTE"
1410 PRINT TAB(10)"THE  DESIRED  PARAMETERS.   THE DATA
     MUST BE CONVERTED USING"
```

Figure A.9 MAIN.BAS (cont.)

```
1420 PRINT TAB(10)"OPTION TWO BEFORE THE PARAMETERS CAN BE
     CALCULATED. YOU WILL"
1430 PRINT TAB(10)"THEN PROVIDE A NAME FOR THE  FILES THAT
     STORE THE CALCULATED"
1440 PRINT TAB(10)"PARAMETERS.  USING  THESE FILE  NAMES,
     THE FIFTH OPTION LETS"
1450 PRINT TAB(10)"YOU EXAMINE THE DATA THAT CAN BE
     PLOTTED."
1455 PRINT:INPUT "PRESS RETURN TO CONTINUE";N
1456 CLS
1460 PRINT TAB(15)"OPTION SIX  MUST BE USED  TO SORT THE
     X VALUES  BEFORE"
1470 PRINT TAB(10)"THEY CAN BE PLOTTED.  THE SEVENTH OPTION
     LETS YOU ADD A GRID"
1480 PRINT TAB(10)"TO THE  PLOT AND ALSO MARK THE DATA
     POINTS.  THE PLOTTING IS"
1490 PRINT TAB(10)"IS  EXECUTED  WITH  THE  EIGHTH OPTION.
     YOU CAN PLOT UP TO"
1500 PRINT TAB(10)"THREE CURVES PER PLOT  AND RECEIVE  A
     HARD  COPY BY PRESSING"
1510 PRINT TAB(10)"THE  CTRL-PRTSC  BUTTONS.   THE FINAL
     OPTION TERMINATES THIS"
1520 PRINT TAB(10)"PROGRAM."
1530 PRINT: INPUT "PRESS RETURN TO CONTINUE";N
2000  '---------------PLOTTER SET-UP----------------
2010 SCREEN 0: WIDTH 80: COLOR 14,0,7:  CLS:  LOCATE 10,1
2020 PRINT "WHICH DISK DRIVE DO YOU WANT TO STORE YOUR
     DATA FILES ON";
2030 PRINT
2040 PRINT "DRIVE A IS EITHER ON YOUR LEFT OR ON TOP"
2050 PRINT "DRIVE B IS EITHER ON YOUR RIGHT OR ON THE
     BOTTOM"
2070 PRINT
2080 INPUT "ENTER [ A OR B ]";FD$: FD$=FD$+":" : PD$="C:"
2090 GRD$="NO GRID": MRK$="MARK   "
2100 OPT$="CURVE"
2110 GOSUB 2230
2120 GOSUB 2400
2130 X%=5+DX%: Y%=12: LOCATE X%,Y%,1,0,7
2140 FIRST%=5: LAST%=13
2150 ANS$=INKEY$: IF ANS$="" THEN 2150
2160 IF ANS$=CHR$(0)+CHR$(80) THEN IF X%<LAST% THEN X%=
     X%+1 ELSE X%=FIRST%
2170 IF ANS$=CHR$(0)+CHR$(72) THEN IF X%>FIRST% THEN X%=
     X%-1 ELSE X%=LAST%
2180 LOCATE X%,Y%:  IF ANS$<>CHR$(13) THEN 2150
2190 FLAG1%=CSRLIN:  FLAG1%=FLAG1%-4
2200 ON FLAG1% GOSUB 3120,2740,2630,2800,2680,3170,2870,
     3550,3680
2210 IF FLAG1%<>1 THEN GOTO 2120 ELSE GOTO 2130
```

Figure A.9 MAIN.BAS (cont.)

```
2220 '--------------INITIALIZING CONSTANTS---------------
2230 DIM X(300),B(300),C(300),D(300),Y(300)
2240 DIM FILESTK$(10)
2250 OUT 985,6
2260 SX%=5:  SY%=55
2270 M$="COMMAND :"
2280 BLANK$="                                          "
2290 B$="             "
2300 SCREEN 0,1: CLS
2310 HDPOS%=48
2320 KEY(1) ON: ON KEY (1) GOSUB 2350
2330 RETURN
2340 '-------------CHANGE FOREGROUND COLORS------------
2350 FG%=FG%+1
2360 IF FG%>15 THEN FG%=1
2370 OUT 985,FG%
2380 RETURN
2390 '---------PRINT MAIN OPTION MENU-------------------
2400 SCREEN 0,1: COLOR 14,0,7: CLS: LOCATE 3,1
2410 PRINT TAB(5) "********       MAIN OPTIONS MENU
     ********"
2420 PRINT
2430 PRINT TAB(5) "*       _  RECORD TUNNNEL DATA
           *"
2440 PRINT TAB(5) "*       _  REDUCE RAW TUNNEL DATA
           *"
2450 PRINT TAB(5) "*       _  EXAMINE DATA FILE FOR TUNNEL
      DATA  *"
2460 PRINT TAB(5) "*       _  CALCULATE Cd, Cl, AND E.F.P.A.
           *"
2470 PRINT TAB(5) "*       _  EXAMINE DATA FILE FOR PLOTTI
     NG     *"
2480 PRINT TAB(5) "*       _  SORT X VALUES
           *"
2490 PRINT TAB(5) "*       _  PLOTTING OPTION MENU
           *"
2500 PRINT TAB(5) "*       _  DO PLOTTING
           *"
2510 PRINT TAB(5) "*       _  EXIT
           *"
2520 PRINT
2530 PRINT TAB(5) "**************************************
     ********"
2540 LOCATE 3,65,1: PRINT "STATUS"
2550 LOCATE 4,55 : PRINT "-----------------------"
2560 LOCATE SX%,SY% : PRINT "DATA FILE DRIVE   = "+FD$
2570 LOCATE SX%+2,SY% : PRINT "USER OPTIONS      :"
2580 LOCATE SX%+3,SY% : PRINT OPT$
2590 LOCATE SX%+4,SY% : PRINT GRD$
2600 LOCATE SX%+5,SY% : PRINT MRK$
```

Figure A.9 MAIN.BAS (cont.)

```
2610 RETURN
2620 '---DATA EDITOR FOR TUNNEL DATA------------
2630 CHAINFILE$ = PD$+"ADATA.BAS"
2640 DX%=2
2650 COMMON FD$,PD$,DX%
2660 CHAIN CHAINFILE$
2670 RETURN
2680 '---DATA EDITOR FOR PLOTTING DATA----------
2690 CHAINFILE$ = PD$+"BDATA.BAS"
2700 DX%=5
2710 COMMON FD$,PD$,DX%
2720 CHAIN CHAINFILE$
2730 RETURN
2740 '----DATA REDUCTION-----------------------
2750 CHAINFILE$ = PD$+"RED.BAS"
2760 DX%=3
2770 COMMON PD$,FD$,DX%
2780 CHAIN CHAINFILE$
2790 RETURN
2800 '-----CALCULATE ROUTINE-------------
2810 CHAINFILE$ = PD$ + "COMP.BAS"
2820 DX%=4
2830 COMMON PD$,FD$,DX%
2840 CHAIN CHAINFILE$
2850 REUTRN
2860 '----PLOTTING OPTION MENU------------
2870  CLS: SCREEN 0,1: LOCATE 6,1
2880 PRINT TAB(20) "******   PLOTTING OPTION MENU     ***
     ***"
2890 PRINT
2900 PRINT TAB(20) "*      _     GRID &    MARK
       *"
2910 PRINT TAB(200 "*      _     GRID & NO MARK
       *"
2920 PRINT TAB(20) "*      _  NO GRID &    MARK
       *"
2930 PRINT TAB(20) "*      _  NO GRID & NO MARK
       *"
2940 PRINT TAB(20) "*      _  EXIT
       *"
2950 PRINT
2960 PRINT TAB(20) "**********************************
     ***"
2970 X%=8: Y%=27: LOCATE X%,Y%,1,0,7
2980 FIRST%=8: LAST%=12
2990 ANS$=INKEY$: IF ANS$="" THEN 2990
3000 IF ANS$=CHR$(0)+CHR$(80) THEN IF X%<LAST% THEN X%=
     X%+1 ELSE X%=FIRST%
3010 IF ANS$=CHR$(0)+CHR$(72) THEN IF X%>FIRST% THEN X%=
     X%+1 ELSE X%=LAST%
```

Figure A.9 MAIN.BAS (cont.)

```
3020 LOCATE X%,Y%: IF ANS$<>CHR$(13) THEN 2990
3030 FLAG%=CSRLIN: FLAG%=FLAG%-7
3040 ON FLAG% GOSUB 3070,3080,3090,3100,3110
3050 DX%=7
3060 RETURN
3070 GRD$="GRID   ": MRK$="MARK   " : RETURN
3080 GRD$="GRID   ": MRK$="NO MARK" : RETURN
3090 GRD$="NO GRID": MRK$="MARK   " : RETURN
3100 GRD$="NO GRID": MRK$="NO MARK" : RETURN
3110 RETURN
3120 '-------RECORD TUNNEL DATA---------------
3130 CHAINFILE$ = PD$ + "RUNS.BAS"
3140 DX%=1
3145 COMMON PD$,FD$,DX%
3150 CHAIN CHAINFILE$
3155 RETURN
3160 '-----SORTING---------------------
3170 CLS: INPUT "ENTER NAME OF FILE TO BE SORTED";FILE$:
     FILE$=FD$+FILE$
3180 PRINT "READING FILE "+FILE$+"...": BEEP
3190 GOSUB 3370
3200 PRINT "SORTING"
3210 L%=2: K%=NOD%-1: R%=NOD%
3220 WHILE (L%<=R%)
3230   FOR J%=R% TO L% STEP -1
3240     IF (X(J%-1)>X(J%)) THEN SWAP X(J%),X(J%-1): SWAP
     Y(J%),Y(J%-1): K%=J%
3250     NEXT
3260   L%=K%-1
3270   FOR J% = L% TO R%
3280     IF (X(J%-1)>X(J%)) THEN SWAP X(J%),X(J%-1): SWAP
     Y(J%),Y(J%-1): K%=J%
3290     NEXT
3300   R%=K%+1
3310   WEND
3320 CLS:BEEP
3330 PRINT "SORTED FILE ";FILE$;" BEING SAVED ...": GOSUB
     3460
3340 DX%=6
3350 RETURN
3360 '-----READING A FILE--------------------
3370 OPEN FILE$ FOR INPUT AS #2
3380 INPUT #2,NOD%,Y1$,X1$,CONFIG$,CONF
3390 INPUT #2,Q
3400   FOR J = 1 TO NOD%
3410       INPUT #2,Y(J),X(J)
3420       NEXT
3430 CLOSE #2
3440 RETURN
3450 '-----SAVE A FILE--------------------
```

Figure A.9 MAIN.BAS (cont.)

```
3460 OPEN FILE$ FOR OUTPUT AS #1
3470 WRITE #1,NOD%,Y1$,X1$,CONFIG$,CONF
3480 WRITE #1,Q
3490   FOR J = 1 TO NOD%
3500       WRITE #1,Y(J),X(J)
3510       NEXT
3520 CLOSE #1
3530 RETURN
3540 '---------INITIALIZING PLOTTER-----------------
3550 CLS: INPUT "NUMBER OF DATA FILE(S) TO BE PLOTTED ON
     THE SAME PLOT =";NOF%
3560 PRINT "ENTER FILE NAME (S) :":BEEP
3570 FOR I% = 1 TO NOF%
3580   PRINT "FILE #";:PRINT USING "##";I%; : INPUT " =";
     FILE$
3590   FILE$ = FD$ + FILE$
3600   FILESTK$(I%) = FILE$
3601   PRINT "HOW DO YOU WANT THIS CURVE PLOTTED ?  ENTER
      (POINT) TO JUST PLOT"
3602   INPUT "THE POINTS OR (CURVE) TO PRODUCE A CURVE
       FIT";OPT$
3610   CURVE$(I%) = OPT$
3620   NEXT
3630 PFILE$ = PD$+"MAIN.BAS": FILE2$= PD$ + "PLOTTER.BAS"
3640 SL% = 100
3650 CHAIN FILE2$,30,ALL
3660 DX%=8
3670 RETURN
3680 CLS:
3690 LOCATE 10,5: PRINT "PROGRAM TERMINATED, REMOVE DATA
     FILES FROM DISK DRIVE"
3700 LOCATE 11,5: PRINT "AND SECURE COPMUTER AND AMPLIFIE
      S"
3710 END
```

Figure A.9 MAIN.BAS (cont.)

```
10 'NAME:  Data Acquisition And Control (DAAC)
20 '       HEADER for BASICA
30 '
40 'FILE NAME:  DACHDR.BAS
50 '
60 'DOS DEVICE NAME:  DAAC
70 '
80 'RESERVED FUNCTION NAMES:
90 '          AINM, AINS, AINSC, AOUM, AOUS,
100 '          BINM, BINS, BITINS, BITOUS, BOUM, BOUS,
110 '          CINM, CINS, CSET, DELAY
120 'RESERVED DEF SEG VALUE NAME:  DSEG
130 '
140 'NAMES DEFINED AND USED BY HEADER:
150 '          ADAPT%, AI, COUNT, FOUND%,
160 '          HNAME$, SG%, STAT%
170 '
180 '
190 'When using the BASICA Interpreter, this header
200 'must be executed before any function calls are
210 'made that access the DAAC adapter.  It initializes
220 'a number of variables for each function call.  These
230 'variables are reserved and should not be used except
240 'to access the DAAC adapter.  This routine also does a
250 'DEF SEG to the segment where the DAAC Device Driver
260 '(DAC.COM) is loaded.  If you execute a DEF SEG to
270 'access other hardware, you must DEF SEG to the segment
280 'of the DAAC Device Driver before any subsequent
290 'calls to access the DAAC adapter.
300 '
310 '
320 FOUND% = 0
330 SG% = &H2E
340 'Start searching the interrupt vectors until you find
350 'one that points to the DAAC device driver.
360 'Do a DEF SEG to that segment.
370 WHILE ((SG% <= &H3E) AND (FOUND% = 0))
380       DEF SEG = 0
390       DSEG = PEEK(SG%) + PEEK(SG% + 1) * 256
400       DEF SEG = DSEG
410       HNAME$=""
420       FOR AI=10 TO 17
430               HNAME$ = HNAME$ + CHR$(PEEK(AI))
440       NEXT AI
450       IF HNAME$ = "DAAC    " AND PEEK(18) + PEEK(19) <>
          0 THEN FOUND% = 1
460       SG% = SG% + 4
470 WEND
480 IF FOUND% = 0 THEN PRINT "ERROR: DEVICE DRIVER DAC.COM
    NOT FOUND" : END
```

Figure A.10 RUNS.BAS - Data Acquisition Program

```
490 'Now initialize all function name variables for calls
500 'to access the device driver.
510 AINM         = PEEK(&H13) * 256 + PEEK(&H12)
520 AINS         = PEEK(&H15) * 256 + PEEK(&H14)
530 AINSC        = PEEK(&H17) * 256 + PEEK(&H16)
540 AOUM         = PEEK(&H19) * 256 + PEEK(&H18)
550 AOUS         = PEEK(&H1B) * 256 + PEEK(&H1A)
560 BINM         = PEEK(&H1D) * 256 + PEEK(&H1C)
570 BINS         = PEEK(&H1F) * 256 + PEEK(&H1E)
580 BITINS       = PEEK(&H21) * 256 + PEEK(&H20)
590 BITOUS       = PEEK(&H23) * 256 + PEEK(&H22)
600 BOUM         = PEEK(&H25) * 256 + PEEK(&H24)
610 BOUS         = PEEK(&H27) * 256 + PEEK(&H26)
620 CINM         = PEEK(&H29) * 256 + PEEK(&H28)
630 CINS         = PEEK(&H2B) * 256 + PEEK(&H2A)
640 CSET         = PEEK(&H2D) * 256 + PEEK(&H2C)
650 DELAY        = PEEK(&H2F) * 256 + PEEK(&H2E)
660 'Finally, execute any call to re-initialize the
670 'device driver from any former invocation of BASIC.
680 ADAPT% = 0
690 COUNT = 1
700 STAT% = 0
710 CALL DELAY (ADAPT%, COUNT, STAT%)
720 '
730 'End of DAAC BASICA Header
740 '
750    REM-- RUNS.BAS;  PROGRAM TO RECORD THE DATA FROM
    WIND
760    REM-- TUNNEL RUNS
770    KEY OFF: COLOR 15,1,4: CLS
780    CLS
790    PRINT: PRINT
800    PRINT TAB(10) "HELO CONFIGURATIONS"
810   PRINT: PRINT
820   PRINT TAB(5) "1.  ATTACK NOSE, STRAIGHT TAIL"
830   PRINT TAB(5) "2.  ATTACK NOSE, LOW TAIL"
840   PRINT TAB(5) "3.  ATTACK NOSE, HIGH TAIL"
850   PRINT TAB(5) "4.  SMOOTH NOSE, STRAIGHT TAIL"
860   PRINT TAB(5) "5.  SMOOTH NOSE, LOW TAIL"
870   PRINT TAB(5) "6.  SMOOTH NOSE, HIGH TAIL"
880   PRINT TAB(5) "7.  BLUNT NOSE, STRAIGHT TAIL"
890   PRINT TAB(5) "8.  BLUNT NOSE, LOW TAIL"
900   PRINT TAB(5) "9.  BLUNT NOSE, HIGH TAIL"
910   PRINT: PRINT
920   INPUT "WHICH CONFIGURATION IS BEING RUN";N
930   PRINT
940   INPUT "DOES CONFIGURATION INCLUDE LANDING GEAR";Z$
950   ON N GOSUB 4550,4620,4690,4750,4820,4870,4920,4990,
      5040
955   '-------  WEIGHTS ARE IN POUNDS--------
```

Figure A.10 RUNS.BAS (cont.)

```
960   IF CONF = 10  THEN W = 22.7892
970   IF CONF = 15  THEN W = 23.4623
980   IF CONF = 20  THEN W = 24.6945
990   IF CONF = 25  THEN W = 25.3676
1000   IF CONF = 30  THEN W = 24.6945
1010   IF CONF = 35  THEN W = 25.3676
1020   IF CONF = 40  THEN W = 20.8134
1030   IF CONF = 45  THEN W = 21.5866
1040   IF CONF = 50  THEN W = 22.7187
1050   IF CONF = 55  THEN W = 23.4913
1060   IF CONF = 60  THEN W = 22.7187
1070   IF CONF = 65  THEN W = 23.4913
1080   IF CONF = 70  THEN W = 21.3315
1090   IF CONF = 75  THEN W = 23.2892
1100   IF CONF = 80  THEN W = 23.2368
1110   IF CONF = 85  THEN W = 25.1945
1120   IF CONF = 90  THEN W = 23.2368
1130   IF CONF = 95  THEN W = 25.1945
1140   CLS
1150   REM-----RECORD TUNNEL DATA-------------------
1160 DIM L(100),D(100),Y(100),PM(100),YM(100),RM(100),AOA(
     100)
1170 DIM DAT(399),DAT%(399),DAT1(399),DAT1%(399)
1180 COLOR 15,1: KEY OFF: CLS
1190 PRINT "RECORDING OF WIND TUNNEL RAW DATA"
1200 INPUT "WHAT IS THE TUNNEL SPEED (Q) FOR THIS RUN";Q
1210 PRINT
2670 PRINT "WITHOUT THE MODEL IN PLACE, ADJUST THE ZERO
     SET SCREWS TO ZERO"
2671 PRINT "OUT EACH AMPLIFIER.  AFTER ALL AMPLIFIERS ARE
     ZEROED, LOAD THE"
2672 PRINT "MODEL ONTO THE STING.  WHEN THE MODEL IS
     MOUNTED, PRESS RETURN"
2673 INPUT "TO RECORD A NO FORCE ZERO READING";N
2690 PRINT:PRINT "  ZD       ZL      ZY        ZPM      ZYM
     ZRM     ZAOA"
2700 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
2710 ADAPT%= 0: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=3
2720 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT%(0),STAT%)
2730 ZD = 0:ZPM=0:ZL=0:ZYM=0
2740 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
2750 FOR J = 0 TO 396 STEP 4
2760 DAT(J)=(DAT%(J)/204.8)-10
2770 ZD =ZD + DAT(J)
2780 NEXT J
2790 ZD = ZD/100
2800 FOR J = 1 TO 397 STEP 4
2810 DAT(J)=(DAT%(J)/204.8)-10
```

Figure A.10 RUNS.BAS (cont.)

```
2820 ZL = ZL + DAT(J)
2830 NEXT J
2840 FOR J = 2 TO 398 STEP 4
2850 DAT(J)=(DAT%(J)/204.8)-10
2860 ZPM = ZPM + DAT(J)
2870 NEXT J
2880 FOR J = 3 TO 399 STEP 4
2890 DAT(J)=(DAT%(J)/204.8)-10
2900 ZYM = ZYM + DAT(J)
2910 NEXT J
2920 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
2930 ADAPT%= 1: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=2
2940 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT1%(0),STAT%)
2950 ZY =0:ZRM=0:ZAOA=0
2960 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
2970 FOR J = 0 TO 297 STEP 3
2980 DAT1(J)=(DAT1%(J)/204.8)-10
2990 ZRM = ZRM + DAT1(J)
3000 NEXT J
3010 ZRM = ZRM/100
3020 FOR J = 1 TO 298 STEP 3
3030 DAT1(J)=(DAT1%(J)/204.8)-10
3040 ZY = ZY + DAT1(J)
3050 NEXT J
3051 FOR J = 2 TO 299 STEP 3
3052 DAT1(J)=(DAT1%(J)/204.8)-10
3053 ZAOA = ZAOA + DAT1(J)
3054 NEXT J
3060 ZL=ZL/100:ZPM=ZPM/100:ZYM=ZYM/100:ZY=ZY/100:ZAOA=ZAOA
     /100
3070 LOCATE 11,1: PRINT USING "+#.###";ZD: LOCATE 11,10:
     PRINT USING "+#.###";ZL
3080 LOCATE 11,19: PRINT USING "+#.###";ZY
3090 LOCATE 11,28: PRINT USING "+#.###";ZPM
3100 LOCATE 11,37: PRINT USING "+#.###";ZYM
3110 LOCATE 11,46: PRINT USING "+#.###";ZRM
3111 LOCATE 11,55: PRINT USING "+#.###";ZAOA
3120 REM
3130 PRINT:PRINT
3140 INPUT "AFTER PLACING ALL CAL SWITCHES TO + SETTING HIT
     RETURN";X
3150 PRINT
3160 PRINT " CALD      CLL     CALY      CALPM      CALYM
     CALRM     CALAOA"
3170 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
3180 ADAPT%= 0: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=3
3190 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT%(0),STAT%)
```

Figure A.10 RUNS.BAS (cont.)

```
3200 CALD=0:CLL=0:CALYM=0:CALPM=0
3210 IF STAT%<> 0,THEN PRINT USING "EXECUTION ERROR ###"
     ;STAT%:END
3220 FOR J = 0 TO 396 STEP 4
3230 DAT(J)=(DAT%(J)/204.8)-10
3240 CALD =CALD + DAT(J)
3250 NEXT J
3260 CALD = CALD/100
3270 FOR J = 1 TO 397 STEP 4
3280 DAT(J)=(DAT%(J)/204.8)-10
3290 CLL = CLL + DAT(J)
3300 NEXT J
3310 FOR J = 2 TO 398 STEP 4
3320 DAT(J)=(DAT%(J)/204.8)-10
3330 CALPM = CALPM + DAT(J)
3340 NEXT J
3350 FOR J = 3 TO 399 STEP 4
3360 DAT(J)=(DAT%(J)/204.8)-10
3370 CALYM = CALYM + DAT(J)
3380 NEXT J
3390 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
3400 ADAPT%= 1: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=2
3410 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT1%(0),STAT%)
3420 CALY =0:CALRM=0:CALAOA=0
3430 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
3440 FOR J = 0 TO 297 STEP 3
3450 DAT1(J)=(DAT1%(J)/204.8)-10
3460 CALRM = CALRM + DAT1(J)
3470 NEXT J
3480 CALRM = CALRM/100
3490 FOR J = 1 TO 298 STEP 3
3500 DAT1(J)=(DAT1%(J)/204.8)-10
3510 CALY = CALY + DAT1(J)
3520 NEXT J
3521 FOR J = 2 TO 299 STEP 3
3522 DAT1(J)=(DAT1%(J)/204.8)-10
3523 CALAOA = CALAOA + DAT1(J)
3524 NEXT J
3530 CLL=CLL/100:CALPM=CALPM/100:CALYM=CALYM/100:CALY=CALY
     /100:CALAOA=CALAOA/100
3540 LOCATE 18,1: PRINT USING "+#.###";CALD
3550 LOCATE 18,10: PRINT USING "+#.###";CLL
3560 LOCATE 18,19: PRINT USING "+#.###";CALY
3570 LOCATE 18,28: PRINT USING "+#.###";CALPM
3580 LOCATE 18,37: PRINT USING "+#.###";CALYM
3590 LOCATE 18,46: PRINT USING "+#.###";CALRM
3591 LOCATE 18,55: PRINT USING "+#.###";CALAOA
3592 INPUT "PRESS RETURN TO CONTINUE";N
```

Figure A.10 RUNS.BAS (cont.)

```
3593 CLS
3600 PRINT: PRINT "REPLACE THE CAL SWITCHES TO THE CENTER
     POSITION"
3610 PRINT "AFTER YOU HAVE GOTTEN THE WIND TUNNEL UP TO
     SPEED AND ARE READY"
3620 PRINT "TO RECORD DATA PRESS RETURN.  ONCE THE HEADINGS
     ARE PRINTED THE"
3630 PRINT "F2 KEY WILL RECORD THE DATA.  THE F1 KEY WILL
     SAVE THE DATA AND "
3640 PRINT "RERUN YOU TO THE MAIN MENU.  YOU CAN TAKE READ
     INGS FOR ANY ANGLE"
3650 PRINT "OF ATTACK BETWEEN +10 AND -8 DEGREES."
3651 PRINT
3652 PRINT "THIS PROGRAM CORRECTS FOR DRIFT IN THE BALANCE
     AND AMPLIFIERS."
3653 PRINT "YOUR FIRST AND LAST DATA POINTS SHOULD BE TAKEN
     AT ZERO ANGLE OF"
3654 PRINT "ATTACK.  ALSO, DO NOT TAKE MORE THAN ONE DATA
     POINT FOR THE SAME "
3655 PRINT "ANGLE OF ATTACK EXCEPT FOR YOUR FIRST AND LAST
     DATA POINT.  YOU"
3656 PRINT "ARE LIMITED TO 100 DATA POINTS.  REMEMBER, EACH
     RUN IS FOR ONE"
3657 PRINT "SPECIFIC 'Q' SETTING"
3660 INPUT "PRESS RETURN TO CONTINUE";N
3670 CLS
3680  PRINT " DRAG      LIFT      YAW       PITCH      YAW
      ROLL       AOA       TUNNEL"
3690  PRINT "                                MOM.      MOM.
      MOM.                 SPEED"
3700 SOAP = 0: N=4
3710 FOR K = 1 TO 100
3720 ON KEY(1) GOSUB 3820      'SET STOP FLAG
3730 ON KEY(2) GOSUB 3850      'RECORD DATA
3740 KEY(1) ON: KEY(2) ON
3750 IF SOAP = 2 THEN GOTO 3780
3760 IF SOAP = 1 THEN GOTO 4370
3770 GOTO 3720
3780 SOAP = 0
3790 NOD% = K
3800 NEXT K
3810 GOTO 4370
3820 REM  SET STOP FLAG
3830 SOAP = 1
3840 RETURN
3850 REM STEPS TO RECORD DATA
3860 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
3870 ADAPT%= 0: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=3
3880 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT%(0),STAT%)
```

Figure A.10 RUNS.BAS (cont.)

```
3890 D(K)=0:L(K)=0:YM(K)=0:PM(K)=0
3900 IF STAT%<> 0,THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
3910 FOR J = 0 TO 396 STEP 4
3920 DAT(J)=(DAT%(J)/204.8)-10
3930 D(K) =D(K) + DAT(J)
3940 NEXT J
3950 D(K) = D(K)/100
3960 FOR J = 1 TO 397 STEP 4
3970 DAT(J)=(DAT%(J)/204.8)-10
3980 L(K) = L(K) + DAT(J)
3990 NEXT J
4000 FOR J = 2 TO 398 STEP 4
4010 DAT(J)=(DAT%(J)/204.8)-10
4020 PM(K) = PM(K) + DAT(J)
4030 NEXT J
4040 FOR J = 3 TO 399 STEP 4
4041 DAT(J)=(DAT%(J)/204.8)-10
4050 YM(K) = YM(K) + DAT(J)
4060 NEXT J
4070 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
4080 ADAPT%= 1: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=2
4090 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT1%(0),STAT%)
4100 Y(K) =0:RM(K)=0: AOA(K)=0
4110 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
4120 FOR J = 0 TO 297 STEP 3
4130 DAT1(J)=(DAT1%(J)/204.8)-10
4140 RM(K) = RM(K) + DAT1(J)
4150 NEXT J
4160 RM(K) = RM(K)/100
4170 FOR J = 1 TO 298 STEP 3
4180 DAT1(J)=(DAT1%(J)/204.8)-10
4190 Y(K) = Y(K) + DAT1(J)
4200 NEXT J
4210 FOR J = 2 TO 299 STEP 3
4220 DAT1(J)=(DAT1%(J)/204.8)-10
4230 AOA(K) = AOA(K) + DAT1(J)
4240 NEXT J
4250 AOA(K)=AOA(K)/100:L(K)=L(K)/100:PM(K)=PM(K)/100:YM(K)
     =YM(K)/100:Y(K)=Y(K)/100
4255 B = AOA(K) - ZAOA
4256 AOA(K) = (5.8469*B) + (.0077583*(B^2))
4260 IF N> 23 THEN N=4:CLS:PRINT " DRAG      LIFT       YAW
     PITCH      YAW      ROLL      AOA         TUNNEL":PRINT "
                MOM.       MOM.     MOM.                SPEED"
4270 LOCATE N,1: PRINT USING "+#.###";D(K)
4280 LOCATE N,10: PRINT USING "+#.###";L(K)
4290 LOCATE N,19: PRINT USING "+#.###";Y(K)
```

Figure A.10 RUNS.BAS (cont.)

```
4300 LOCATE N,28: PRINT USING "+#.###";PM(K)
4310 LOCATE N,37: PRINT USING "+#.###";YM(K)
4320 LOCATE N,46: PRINT USING "+#.###";RM(K)
4330 LOCATE N,55: PRINT USING "+#.###";AOA(K)
4340 N=N+1
4350 SOAP = 2
4360 RETURN
4370 SOAP = 0
4371 '---CORRECT FOR DRIFT-------------
4372 DIFFL =(L(NOD%)-L(1)): DIFFD=(D(NOD%)-D(1)):DIFFY=(Y(
     NOD%)-Y(1))
4373 DIFFPM=(PM(NOD%)-PM(1)):DIFFYM=(YM(NOD%)-YM(1)):DIFFR
     M=(RM(NOD%)-RM(1)):DIFFAOA=(AOA(NOD%)-AOA(1))
4374 D=NOD%-1
4375 CORL=DIFFL/D:CORD=DIFFD/D:CORY=DIFFY/D:CORPM=DIFFPM/D
     :CORYM=DIFFYM/D:CORRM=DIFFRM/D:CORAOA=DIFFAOA/D
4376 A=1
4377 FOR K = 2 TO NOD%
4378   L(K)=L(K)-(A*CORL)
4379   D(K)=D(K)-(A*CORD)
4380   Y(K)=Y(K)-(A*CORY)
4381   PM(K)=PM(K)-(A*CORPM)
4382   YM(K)=YM(K)-(A*CORYM)
4383   RM(K)=RM(K)-(A*CORRM)
4385   A = A+1
4386 NEXT K
4387 '------SAVE DATA TO FILE-------------
4390 CLS: PRINT
4400 INPUT "WHAT IS THE NAME FOR THE FILE TO STORE THE RAW
     DATA";FILE$
4410 FILE$=FD$+FILE$
4420 OPEN FILE$ FOR OUTPUT AS #1
4421 WRITE #1,NOD%,CONFIG$,CONF
4430 WRITE #1,Q,W
4470 WRITE #1,ZL,ZD,ZY,ZPM,ZYM,ZRM,ZAOA
4480 WRITE #1,CLL,CALD,CALY,CALPM,CALYM,CALRM,CALAOA
4490    FOR J=1 TO NOD%
4500      WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),AOA(J)
4510      NEXT J
4520 CLOSE #1
4521 CLS:PRINT "DO YOU WANT TO TAKE ANOTHER SET OF DATA
     POINTS FOR THE SAME"
4522 INPUT "CONFIGURATION BUT A DIFFERENT 'Q' SETTING (Y/N
     )";ANS$
4523 IF ANS$="Y" OR ANS$="y" THEN GOTO 4524 ELSE GOTO 4530
4524 PRINT: INPUT "WHAT IS THE 'Q' VALUE FOR THIS RUN";Q
4525 GOTO 3610
4530 '----RETURN TO MAIN PROGRAM
4531 COMMON PD$,FD$,DX%
4532 CHAIN PD$+"MAIN.BAS",2110
```

Figure A.10 RUNS.BAS (cont.)

```
4540   REM----SUBROUTINES FOR CONFIGURATION MARKING-------
4550   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 4590
4560   CONF = 10
4570   CONFIG$="ATTACK NOSE/STRAIGHT TAIL WITHOUT GEAR"
4580   GOTO 4610
4590   CONF = 15
4600   CINFIG$="ATTACK NOSE/STRAIGHT TAIL WITH GEAR"
4610   RETURN
4620   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 4660
4630   CONF = 20
4640   CONFIG$="ATTACK NOSE/LOW TAIL WITHOUT GEAR"
4650   GOTO 4680
4660   CONF = 25
4670   CINFIG$="ATTACK NOSE/LOW TAIL WITH GEAR"
4680   RETURN
4690   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 4730
4700   CONF = 30
4710   CONFIG$="ATTACK NOSE/HIGH TAIL WITHOUT GEAR"
4720   GOTO 4740
4730   CONF = 35: CINFIG$="ATTACK NOSE/HIGH TAIL WITH GEAR"
4740   RETURN
4750   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 4790
4760   CONF = 40
4770   CONFIG$="SMOOTH NOSE/STRAIGHT TAIL WITHOUT GEAR"
4780   GOTO 4810
4790   CONF = 45
4800   CINFIG$="SMOOTH NOSE/STRAIGHT TAIL WITH GEAR"
4810   RETURN
4820   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 4850
4830   CONF=50:CONFIG$="SMOOTH NOSE/LOW TAIL WITHOUT GEAR"
4840   GOTO 4860
4850   CONF = 55: CINFIG$="SMOOTH NOSE/LOW TAIL WITH GEAR"
4860   RETURN
4870   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 4900
4880   CONF=60:CONFIG$="SMOOTH NOSE/HIGH TAIL WITHOUT GEAR"
4890   GOTO 4910
4900   CONF = 65: CINFIG$="SMOOTH NOSE/HIGH TAIL WITH GEAR"
4910   RETURN
4920   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 4960
4930   CONF = 70
4940   CONFIG$="ATTACK NOSE/STRAIGHT TAIL WITHOUT GEAR"
4950   GOTO 4980
4960   CONF = 75
4970   CINFIG$="ATTACK NOSE/STRAIGHT TAIL WITH GEAR"
4980   RETURN
4990   IF Z$ = "Y" OR Z$ = "y" THEN GOTO 5020
5000   CONF = 80:CONFIG$="BLUNT NOSE/LOW TAIL WITHOUT GEAR"
5010   GOTO 5030
5020   CONF = 85: CINFIG$="BLUNT NOSE/LOW TAIL WITH GEAR"
5030   RETURN
```

Figure A.10 RUNS.BAS (cont.)

```
5040    IF Z$ = "Y" OR Z$ = "y" THEN GOTO 5070
5050    CONF=90:CONFIG$="BLUNT NOSE/HIGH TAIL WITHOUT GEAR"
5060    GOTO 5080
5070    CONF = 95: CINFIG$="SMOOT NOSE/HIGH TAIL WITH GEAR"
5080    RETURN
```

Figure A.10 RUNS.BAS (cont.)

```
10   REM----RED.BAS    (CONVERT RAW COUNTS TO FORCES)----
20   REM
30   REM
40   COLOR 15,1:KEY OFF: CLS
45   DIM L(100),L1(100),L2(100),D(100),D1(100),D2(100),
      (100),Y1(100),Y2(100)
46   DIM PM(100),PM1(100),PM2(100),YM(100),YM1(100),YM2(10
     0),RM(100),RM1(100),RM2(100)
47   DIM LF(100),DF(100),AOA(100)
50   PRINT:PRINT"WHAT IS THE NAME OF THE FILE THAT CONTAINS
     THE RAW DATA"
60   INPUT"THAT YOU WISH TO CONVERT";FILE$
70   FILE$=FD$+FILE$
80   '-----READ IN RAW DATA FILE-----
90   OPEN FILE$ FOR INPUT AS #1
100  INPUT #1,NOD%,CONFIG$,CONF
110  INPUT #1,Q,W
150  INPUT #1,ZL,ZD,ZY,ZPM,ZYM,ZRM,ZAOA
160  INPUT #1,CLL,CALD,CALY,CALPM,CALYM,CALRM,CALAOA
170     FOR J = 1 TO NOD%
180     INPUT #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),AOA(J)
190     NEXT J
200  CLOSE #1
210  PRINT:PRINT"PLEASE WAIT WHILE PERFORMING CONVERSION"
211  PRINT "THE CONVERSION WILL TAKE APPROXIMATELY TEN MIN
     UTES FOR "
212  PRINT "TWENTY DATA POINTS"
220  '------READ IN CALIBRATION CONSTANTS--------
221  OPEN "C:CONST" FOR INPUT AS #1
222  INPUT #1,INCALL,INCALD,INCALY,INCALPM,INCALYM,INCALRM
240  INPUT #1,K1LPOS,K2LPOS,K1DPOS,K2DPOS,K1YPOS,K2YPOS
250  INPUT #1,K1PMPOS,K2PMPOS,K1YMPOS,K2YMPOS,K1RMPOS,
     K2RMPOS
260  INPUT #1,K1LNEG,K2LNEG,K1DNEG,K2DNEG,K1YNEG,K2YNEG
270  INPUT #1,K1PMNEG,K2PMNEG,K1YMNEG,K2YMNEG,K1RMNEG,
     K2RMNEG
280  INPUT #1,DDDL1P,DDDL2P,DYDL1P,DYDL2P,DPMDL1P,DPMDL2P,
     DYMDL1P,DYMDL2P
290  INPUT #1,DRMDL1P,DRMDL2P
300  INPUT #1,DLDD1P,DLDD2P,DYDD1P,DYDD2P,DPMDD1P,DPMDD2P,
     DYMDD1P,DYMDD2P
310  INPUT #1,DRMDD1P,DRMDD2P
320  INPUT #1,DLDY1P,DLDY2P,DDDY1P,DDDY2P,DPMDY1P,DPMDY2P,
     DYMDY1P,DYMDY2P
330  INPUT #1,DRMDY1P,DRMDY2P
340  INPUT #1,DLDPM1P,DLDPM2P,DDDPM1P,DDDPM2P,DYDPM1P,DYDP
     M2P,DYMDPM1P,DYMDPM2P
350  INPUT #1,DRMDPM1P,DRMDPM2P
360  INPUT #1,DLDYM1P,DLDYM2P,DDDYM1P,DDDYM2P,DYDYM1P,DYDY
     M2P,DPMDYM1P,DPMDYM2P
```

Figure A.11 RED.BAS - Data Reduction Program

```
370  INPUT #1,DRMDYM1P,DRMDYM2P
380  INPUT #1,DLDRM1P,DLDRM2P,DDDRM1P,DDDRM2P,DYDRM1P,DYDR
     M2P,DPMDRM1P,DPMDRM2P
390  INPUT #1,DYMDRM1P,DYMDRM2P
400  INPUT #1,DDDL1N,DDDL2N,DYDL1N,DYDL2N,DPMDL1N,DPMDL2N,
     DYMDL1N,DYMDL2N
410  INPUT #1,DRMDL1N,DRMDL2N
420  INPUT #1,DLDD1N,DLDD2N,DYDD1N,DYDD2N,DPMDD1N,DPMDD2N,
     DYMDD1N,DYMDD2N
430  INPUT #1,DRMDD1N,DRMDD2N
440  INPUT #1,DLDY1N,DLDY2N,DDDY1N,DDDY2N,DPMDY1N,DPMDY2N,
     DYMDY1N,DYMDY2N
450  INPUT #1,DRMDY1N,DRMDY2N
460  INPUT #1,DLDPM1N,DLDPM2N,DDDPM1N,DDDPM2N,DYDPM1N,DYDP
     M2N,DYMDPM1N
470  INPUT #1,DYMDPM2N,DRMDPM1N,DRMDPM2N
480  INPUT #1,DLDYM1N,DLDYM2N,DDDYM1N,DDDYM2N,DYDYM1N,DYDY
     M2N,DPMDYM1N
490  INPUT #1,DPMDYM2N,DRMDYM1N,DRMDYM2N
500  INPUT #1,DLDRM1N,DLDRM2N,DDDRM1N,DDDRM2N,DYDRM1N,DYDR
     M2N,DPMDRM1N
510  INPUT #1,DPMDRM2N,DYMDRM1N,DYMDRM2N
520  CLOSE #1
530  '-----CONVERT RAW COUNTS TO FORCES------
540  FOR J = 1 TO NOD%
550  A = ((INCALL/(CLL-ZL))*(L(J)-ZL))
560  B = ((INCALD/(CALD-ZD))*(D(J)-ZD))
570  C = ((INCALY/(CALY-ZY))*(Y(J)-ZY))
580  D=((INCALPM/(CALPM-ZPM))*(PM(J)-ZPM))
590  E=((INCALYM/(CALYM-ZYM))*(YM(J)-ZYM))
600  F=((INCALRM/(CALRM-ZRM))*(RM(J)-ZRM))
610       IF A <= 0 THEN K1=K1LNEG:K2=K2LNEG  ELSE K1=K1LPOS
     :K2=K2LPOS
615       L(J) = (K1*A) + (K2*A^2)
620       IF B <= 0 THEN K1=K1DNEG:K2=K2DNEG  ELSE K1=K1DPOS
     :K2=K2DPOS
625       D(J) = (K1*B) + (K2*B^2)
630       IF C <= 0 THEN K1=K1YNEG:K2=K2YNEG  ELSE K1=K1YPOS
     :K2=K2YPOS
635       Y(J) = (K1*C) + (K2*C^2)
640       IF D <= 0 THEN K1=K1PMNEG:K2=K2PMNEG   ELSE K1=K1
     PMPOS:K2=K2PMPOS
645       PM(J) = (K1*D) + (K2*D^2)
650       IF E <= 0 THEN K1=K1YMNEG:K2=K2YMNEG   ELSE K1=K1
     YMPOS:K2=K2YMPOS
655       YM(J) = (K1*E) + (K2*E^2)
660       IF F <= 0 THEN K1=K1RMNEG:K2=K2RMNEG   ELSE K1=K1
     RMPOS:K2=K2RMPOS
665       RM(J) = (K1*F) + (K2*F^2)
670  NEXT J
```

Figure A.11 RED.BAS (cont.)

```
680  '-----PERFORM INTERACTON CORRECTIONS----
690  FOR K = 1 TO NOD%
695  '------FIRST CONSTANT DETERMINATION----------
700    IF L(K) > 0 THEN GOTO 710 ELSE GOTO 720
710    DD1=DDDL1P:DD6=DDDL2P:DY1=DYDL1P:DY6=DYDL2P:DPM1=DP
       MDL1P:DPM6=DPMDL2P:DYM1=DYMDL1P:DYM6=DYMDL2P:DRM1=D
       RMDL1P:DRM6=DRMDL2P
715    GOTO 730
720    DD1=DDDL1N:DD6=DDDL2N:DY1=DYDL1N:DY6=DYDL2N:DPM1=DP
       MDL1N:DPM6=DPMDL2N:DYM1=DYMDL1N:DYM6=DYMDL2N:DRM1=D
       RMDL1N:DRM6=DRMDL2N
730    IF D(K) > 0 THEN GOTO 740 ELSE GOTO 750
740    DL1=DLDD1P:DL6=DLDD2P:DY2=DYDD1P:DY7=DYDD2P:DPM2=DP
       MDD1P:DPM7=DPMDD2P:DYM2=DYMDD1P:DYM7=DYMDD2P:DRM2=D
       RMDD1P:DRM7=DRMDD2P
745    GOTO 760
750    DL1=DLDD1N:DL6=DLDD2N:DY2=DYDD1N:DY7=DYDD2N:DPM2=DP
       MDD1N:DPM7=DPMDD2N:DYM2=DYMDD1N:DYM7=DYMDD2N:DRM2=D
       RMDD1N:DRM7=DRMDD2N
760    IF Y(K) > 0 THEN GOTO 770 ELSE GOTO 780
770    DL2=DLDY1P:DL7=DLDY2P:DD2=DDDY1P:DD7=DDDY2P:DPM3=DP
       MDY1P:DPM8=DPMDY2P:DYM3=DYMDY1P:DYM8=DYMDY2P:DRM3=D
       RMDY1P:DRM8=DRMDY2P
775    GOTO 790
780    DL2=DLDY1N:DL7=DLDY2N:DD2=DDDY1N:DD7=DDDY2N:DPM3=DP
       MDY1N:DPM8=DPMDY2N:DYM3=DYMDY1N:DYM8=DYMDY2N:DRM3=D
       RMDY1N:DRM8=DRMDY2N
790    IF PM(K) > 0 THEN GOTO 800 ELSE GOTO 810
800    DL3=DLDPM1P:DL8=DLDPM1P:DD3=DDDPM1P:DD8=DDDPM2P:DY3
       =DYDPM1P:DY8=DYDPM2P:DYM4=DYMDPM1P:DYM9=DYMDPM2P:DR
       M4=DRMDPM1P:DRM9=DRMDPM2P
805    GOTO 820
810    DL3=DLDPM1N:DL8=DLDPM1N:DD3=DDDPM1N:DD8=DDDPM2N:DY3
       =DYDPM1N:DY8=DYDPM2N:DYM4=DYMDPM1N:DYM9=DYMDPM2N:DR
       M4=DRMDPM1N:DRM9=DRMDPM2N
820    IF YM(K) > 0 THEN GOTO 830 ELSE GOTO 840
830    DL4=DLDYM1P:DL9=DLDYM2P:DD4=DDDYM1P:DD9=DDDYM2P:DY4
       =DYDYM1P:DY9=DYDYM2P:DPM4=DPMDYM1P:DPM9=DPMDYM2P:DR
       M5=DRMDYM1P:DRM10=DRMDYM2P
835    GOTO 850
840    DL4=DLDYM1N:DL9=DLDYM2N:DD4=DDDYM1N:DD9=DDDYM2N:DY4
       =DYDYM1N:DY9=DYDYM2N:DPM4=DPMDYM1N:DPM9=DPMDYM2N:DR
       M5=DRMDYM1N:DRM10=DRMDYM2N
850    IF RM(K) > 0 THEN GOTO 860 ELSE GOTO 870
860    DL5=DLDRM1P:DL10=DLDRM2P:DD5=DDDRM1P:DD10=DDDRM2P:D
       Y5=DYDRM1P:DY10=DYDRM2P:DPM5=DPMDRM1P:DPM10=DPMDRM2
       P:DYM5=DYMDRM1P:DYM10=DYMDRM2P
865    GOTO 880
870    DL5=DLDRM1N:DL10=DLDRM2N:DD5=DDDRM1N:DD10=DDDRM2N:D
       Y5=DYDRM1N:DY10=DYDRM2N:DPM5=DPMDRM1N:DPM10=DPMDRM2
```

Figure A.11 RED.BAS (cont.)

```
      N:DYM5=DYMDRM1N:DYM10=DYMDRM2N
880 '-----FIRST INTERACTION CALCULATION---------
890     L1(K)=L(K)-(DL1*D(K))-(DL6*(D(K)^2))
900     D1(K)=D(K)-(DD1*L(K))-(DD6*(D(K)^2))
950 '-----SECOND CONSTANT DETERMINATION-----------
960     IF L1(K) > 0 THEN GOTO 970 ELSE GOTO 980
970     DD1=DDDL1P:DD6=DDDL2P:DY1=DYDL1P:DY6=DYDL2P:DPM1=DP
        MDL1P:DPM6=DPMDL2P:DYM1=DYMDL1P:DYM6=DYMDL2P:DRM1=D
        RMDL1P:DRM6=DRMDL2P
975     GOTO 990
980     DD1=DDDL1N:DD6=DDDL2N:DY1=DYDL1N:DY6=DYDL2N:DPM1=DP
        MDL1N:DPM6=DPMDL2N:DYM1=DYMDL1N:DYM6=DYMDL2N:DRM1=D
        RMDL1N:DRM6=DRMDL2N
990     IF D1(K) > 0 THEN GOTO 1000 ELSE GOTO 1010
1000    DL1=DLDD1P:DL6=DLDD2P:DY2=DYDD1P:DY7=DYDD2P:DPM2=DP
        MDD1P:DPM7=DPMDD2P:DYM2=DYMDD1P:DYM7=DYMDD2P:DRM2=D
        RMDD1P:DRM7=DRMDD2P
1005    GOTO 1020
1010    DL1=DLDD1N:DL6=DLDD2N:DY2=DYDD1N:DY7=DYDD2N:DPM2=DP
        MDD1N:DPM7=DPMDD2N:DYM2=DYMDD1N:DYM7=DYMDD2N:DRM2=D
        RMDD1N:DRM7=DRMDD2N
1020    IF Y1(K) > 0 THEN GOTO 1030 ELSE GOTO 1040
1030    DL2=DLDY1P:DL7=DLDY2P:DD2=DDDY1P:DD7=DDDY2P:DPM3=DP
        MDY1P:DPM8=DPMDY2P:DYM3=DYMDY1P:DYM8=DYMDY2P:DRM3=D
        RMDY1P:DRM8=DRMDY2P
1035    GOTO 1050
1040    DL2=DLDY1N:DL7=DLDY2N:DD2=DDDY1N:DD7=DDDY2N:DPM3=DP
        MDY1N:DPM8=DPMDY2N:DYM3=DYMDY1N:DYM8=DYMDY2N:DRM3=D
        RMDY1N:DRM8=DRMDY2N
1050    IF PM1(K) > 0 THEN GOTO 1060 ELSE GOTO 1080
1060    DL3=DLDPM1P:DL8=DLDPM1P:DD3=DDDPM1P:DD8=DDDPM2P:DY3
        =DYDPM1P:DY8=DYDPM2P:DYM4=DYMDPM1P:DYM9=DYMDPM2P:DR
        M4=DRMDPM1P:DRM9=DRMDPM2P
1070    GOTO 1090
1080    DL3=DLDPM1N:DL8=DLDPM1N:DD3=DDDPM1N:DD8=DDDPM2N:DY3
        =DYDPM1N:DY8=DYDPM2N:DYM4=DYMDPM1N:DYM9=DYMDPM2N:DR
        M4=DRMDPM1N:DRM9=DRMDPM2N
1090    IF YM1(K) > 0 THEN GOTO 1100 ELSE GOTO 1110
1100    DL4=DLDYM1P:DL9=DLDYM2P:DD4=DDDYM1P:DD9=DDDYM2P:DY4
        =DYDYM1P:DY9=DYDYM2P:DPM4=DPMDYM1P:DPM9=DPMDYM2P:DR
        M5=DRMDYM1P:DRM10=DRMDYM2P
1105    GOTO 1120
1110    DL4=DLDYM1N:DL9=DLDYM2N:DD4=DDDYM1N:DD9=DDDYM2N:DY4
        =DYDYM1N:DY9=DYDYM2N:DPM4=DPMDYM1N:DPM9=DPMDYM2N:DR
        M5=DRMDYM1N:DRM10=DRMDYM2N
1120    IF RM1(K) > 0 THEN GOTO 1130 ELSE GOTO 1140
1130    DL5=DLDRM1P:DL10=DLDRM2P:DD5=DDDRM1P:DD10=DDDRM2P:D
        Y5=DYDRM1P:DY10=DYDRM2P:DPM5=DPMDRM1P:DPM10=DPMDRM2
        P:DYM5=DYMDRM1P:DYM10=DYMDRM2P
1135    GOTO 1150
```

Figure A.11 RED.BAS (cont.)

```
1140   DL5=DLDRM1N:DL10=DLDRM2N:DD5=DDDRM1N:DD10=DDDRM2N:D
       Y5=DYDRM1N:DY10=DYDRM2N:DPM5=DPMDRM1N:DPM10=DPMDRM2
       N:DYM5=DYMDRM1N:DYM10=DYMDRM2N
1150 '-----SECOND INTERACTION CALCULATION----------
1160   L2(K)=L(K)-(DL1*D1(K))-(DL6*(D1(K)^2))
1170   D2(K)=D(K)-(DD1*L1(K))-(DD6*(D1(K)^2))
1220 '----COMPARE INTERACTION CALCULATIONS------
1230   DFL=ABS(L1(K)-L2(K)):DFD=ABS(D1(K)-D2(K))
1250   IF DFL < .0005 AND DFD < .0005 THEN GOTO 1330
1260   IF DFL > .0005 THEN L1(K)=L2(K)
1270   IF DFD > .0005 THEN D1(K)=D2(K)
1320   GOTO 960
1330   L(K)=L2(K):D(K)=D2(K)
1340   NEXT K
2330 '----PERFORM WEIGHT TARE CORRECTIONS--------
2335 '--DRAG AND LIFT NEED TO BE CORRECTED FOR AXIS ORIENT
     ATION--
2340 FOR K = 1 TO NOD%
2350   ALPHA = AOA(K)*(3.141593/180) 'CONVERT TO RADIANS
2360   DF(K)=(((-1*D(K))-(W*SIN(ALPHA)))*COS(ALPHA)) + (((
       -1*L(K))-(W*COS(ALPHA)))*SIN(ALPHA))
2370   LF(K)=(((-1*L(K))-(W*COS(ALPHA)))*COS(ALPHA)) - (((
       -1*L(K))-(W*SIN(ALPHA)))*SIN(ALPHA))
2380   NEXT K
2390 '----SAVE REDUCED DATA TO FILE-----------
2395 CLS:PRINT "DATA REDUCTION COMPLETE, SAVING DATA TO ";
     FILE$
2400 OPEN FILE$ FOR OUTPUT AS #2
2410 WRITE #2,NOD%,CONFIG$,CONF
2420 WRITE #2,Q,W
2421 WRITE #2,ZL,ZD,ZY,ZPM,ZYM,ZRM,ZAOA
2422 WRITE #2,CLL,CALD,CALY,CALPM,CALYM,CALRM,CALAOA
2430    FOR J = 1 TO NOD%
2450    WRITE #2,LF(J),DF(J),Y(J),PM(J),YM(J),RM(J),AOA(J)
2460    NEXT J
2470 '---------RETURN TO MAIN PROGRAM---------------
2480 COMMON PD$,FD$,DX%
2490 CHAIN PD$+"MAIN.BAS",2110
2500 '-------END OF REDUCTION----------------------
```

Figure A.11 RED.BAS (cont.)

```
10   '--------DATA EDITOR---------
20   GOSUB 180
30   COLOR 15,1,7: KEY OFF: CLS
40   LOCATE 5,34: PRINT "*OPTIONS MENU*"
50   PRINT
60   PRINT
70   PRINT TAB(30) "1.  CREATE A NEW FILE"
80   PRINT
90   PRINT TAB(30) "2.  EDIT EXISTING FILE"
100   PRINT
110  PRINT TAB(30) "3.  INCREASE NUMBER OF TEST POINTS"
120  PRINT TAB(30) "    IN AN EXISTING FILE"
130  PRINT
140  PRINT TAB(30) "4.  EXIT DATA EDITOR"
150  LOCATE 20,5: INPUT "ENTER YOUR CHOICE (1,2,3,4)";REP%
160  ON REP% GOSUB 390,280,2530,480
170  GOTO 30
180  '---------INITIALIZING CONSTANTS---------------
190  OPTION BASE 1: KEY OFF
200  DIM LF(100),DF(100),YF(100),PM(100),RM(100)
210  DIM YM(100),AOA(100),COMMAND(10)
220  BLANK2$ = "                              "
230  BLANK1$ = "         "
240  BLANK$  = "                  "
250  COMMAND$(1) = "c": COMMAND$(2) = "r": COMMAND$(3)
     = "d"
260  COMMAND$(4) = "e": COMMAND$(5) = "s": COMMAND$(6) =
     "u": COMMAND$(7) = "q"
270  RETURN
280  '-----------EDIT A FILE----------------------
290  CLS: COLOR 15,1,7
300  LOCATE 5,10: INPUT "ENTER YOUR FILE NAME";FILE$
305  FILE$ = FD$+FILE$
310  GOSUB 2680    ' READ IN FILE
320  PAGE% = 1: COL% = 1: SAVED% = 0: OLDXPOS% = 0:
     OLDYPOS% = 0
330  GOSUB 1050  'PRINT BACKGROUND LINES
340  ROW% = 3: GOSUB 1740 :  GOSUB 1900     'HIGHLIGHT
     FIRST ROW AND COLUMN
350  LOCATE 23,30: PRINT "LISTING": BEEP
360  GOSUB 1430  'PRINT COLUMNS 1-7 AND Q VALUE
370  LOCATE 23,20: PRINT BLANK2$
380  GOTO 470
390  '----------CREATE A FILE---------------------
400  COLOR 15,1,7: CLS
410  LOCATE 5,10: INPUT "ENTER YOUR FILE NAME";FILE$
415  FILE$ = FD$ + FILE$
420  LOCATE 7,10: INPUT "ENTER TEST CONFIGURATION";CONFIG$
430  LOCATE 9,10: INPUT "ENTER THE NUMBER OF TEST POINTS
     PER RUN";NOD%
```

Figure A.12 ADATA.BAS - On-Screen Data Editor

```
440  PAGE% = 1: COL% = 1: SAVED% = 0: OLDXPOS% = 0:
     OLDYPOS% = 0
450  GOSUB 1050  'PRINT BACKGROUND LINES
460  ROW% = 3: GOSUB 1740: GOSUB 1900      'HIGHLIGHT
     FIRST ROW AND COLUMN
470  GOSUB 490  ' INSERT NEW DATA
480  COMMON FD$,PD$,DX%
481  CHAIN PD$+"MAIN.BAS",2110
482  RETURN
490  '----------CHANGE OR INSERT DATA--------------
500  LOCATE 23,12: PRINT BLANK$
510  LOCATE 23,12: INPUT REP$: IF REP$ = "" THEN GOTO 530
520  GOTO 570
530  LOCATE 23,12: PRINT BLANK$: LOCATE 23,12: PRINT
     "INVALID COMMAND": BEEP
540   FOR I = 1 TO 500 STEP 1
550   NEXT I
560  GOTO 500
570  REP2$ = RIGHT$(REP$,1)
580  T1% = ASC(REP2$)
590  IF T1% >= 43 AND T1% <= 57 THEN GOTO 650
600  T1% = T1% OR 32: REP2$ = CHR$(T1%)
610  GOSUB 2000
620  IF VALID% = 0 THEN GOTO 500 ELSE VALID% = 0
630  ON FLAG% GOSUB 1780,1670,2080,2340,2770,2170,2260
640  IF REP2$ <> "e" THEN GOTO 500 ELSE GOTO 1040
650  WHILE REP2$ <> "e"
660     NEWDATA = VAL(REP$)
670     IF QUIT% = 1 OR OLDXPOS% = 0 THEN QUIT% = 0: GOTO
      700   ELSE
680           LOCATE OLDXPOS%,OLDYPOS%,0
690           PRINT USING "####.##";OLDDATA
700     LOCATE 23,12: PRINT BLANK$
710     R% = ROW% - 2
720     IF COL% = 1 THEN LF(R%+18*(PAGE%-1))= NEWDATA:
      YPOS% = 7: GOTO 790: ELSE
730     IF COL% = 2 THEN DF(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 18: GOTO 790: ELSE
740     IF COL% = 3 THEN YF(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 29: GOTO 790: ELSE
750     IF COL% = 4 THEN PM(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 40: GOTO 790: ELSE
760     IF COL% = 5 THEN RM(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 51: GOTO 790: ELS
770     IF COL% = 6 THEN YM(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 62: GOTO 790: ELS
780     IF COL% =7 THEN AOA(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 71: GOTO 790: ELS
790     XPOS% = ROW%
800     LOCATE XPOS%,YPOS%: COLOR 0,10
```

Figure A.12 ADATA.BAS (cont.)

```
810      PRINT USING "####.##"; NEWDATA: MODIFIED% = 1
820      SAVED% = 0: COLOR 15,1,7
830      OLDYPOS% = YPOS%: OLDXPOS% = XPOS%: OLDDATA =
         NEWDATA
840      FWD% = 1
850      I% = ROW%+1: GOSUB 1690  'HIGHLIGHT NEW ROW
860      IF I% > NOD%+2 THEN FWD% = 1 ELSE FWD% = 0
870      IF I% > 20 THEN FWD% = 1 ELSE FWD% = 0
880      GOSUB 1800  'HIGHLIGHT NEW ROW
890      COLOR 15,1,7
900      LOCATE 23,12: PRINT BLANK$
910      LOCATE 23,12:INPUT REP$: IF REP$ = "" THEN GOTO 930
920      GOTO 950
930      LOCATE 23,12: PRINT BLANK$: LOCATE 23,12: PRINT
         "INVALID COMMAND": BEEP
940      GOTO 900
950      REP2$ = RIGHT$(REP$,1)
960      T1% = ASC(REP2$)
970      IF T1% >= 43 AND T1% <= 57 THEN GOTO 650
980      T1% = T1% OR 32: REP2$ = CHR$(T1%)
990      GOSUB 2000  'CHECK COMMANDS
1000      IF VALID% = 0 THEN GOTO 500 ELSE VALID% = 0
1010      ON FLAG% GOSUB 1780,1670,2080,2340,2770,2170,2260
          'COMMAND SUBROUTINES
1020      IF REP2$ <> "e" THEN GOTO 900 ELSE GOTO 1030
1030      WEND
1040  RETURN
1050 '-------------PRINT BACKGROUND LINES-----------
1060 COLOR 15,1,7
1070 CLS
1080 LOCATE 2,1
1090 FOR I% = 1 TO 80 : PRINT CHR$(220);: NEXT
1100 LOCATE 1,9: PRINT "LIFT"
1110 LOCATE 1,20: PRINT "DRAG"
1120 LOCATE 1,32: PRINT "YAW"
1130 LOCATE 1,41: PRINT "PITCH"
1140 LOCATE 1,53: PRINT "ROLL"
1150 LOCATE 1,62: PRINT "YAW M."
1160 LOCATE 1,74: PRINT "AOA"
1170 LOCATE 21,1
1180 FOR I% = 1 TO 80 : PRINT CHR$(220);: NEXT
1190 FOR I% = 1 TO 22
1200      LOCATE I%,1: PRINT CHR$(222)
1210      LOCATE I%,80: PRINT CHR$(222)
1220 NEXT
1230 FOR I% = 1 TO 21
1240      LOCATE I%,14: PRINT CHR$(179)
1250      LOCATE I%,25: PRINT CHR$(179)
1260      LOCATE I%,36: PRINT CHR$(179)
1270      LOCATE I%,47: PRINT CHR$(179)
```

Figure A.12 ADATA.BAS (cont.)

```
1280      LOCATE I%,58: PRINT CHR$(179)
1290      LOCATE I%,69: PRINT CHR$(179)
1300 NEXT
1310 FOR I% = 1 TO 18
1320     LOCATE I%+2,2
1330     PRINT USING "###";I%+18*(PAGE%-1);: PRINT ":"
1340     NEXT
1350 COLOR 15,4
1360 LOCATE 22,15: PRINT "C COLUMN  R ROW  D DOWN  E EXIT"
1370 LOCATE 22,46: PRINT "  S SAVE    U UP   Q TUNNEL
     SPEED"
1380 COLOR 14,0: LOCATE 23,44: PRINT CONFIG$
1390 COLOR 15,1,7
1400 LOCATE 22,2: PRINT "[Q =": LOCATE 22,12: PRINT "]"
1410 LOCATE 23,3: PRINT "COMMAND :"
1420 RETURN
1430 '---------PRINT COLUMNS 1-7 AND Q VALUE---------
1440 FOR I% = 1 TO 18
1450    LOCATE I%+2,7: PRINT USING "####.##";LF(I%+18*(PAG
        E%-1))
1460    NEXT
1470 FOR I% = 1 TO 18
1480    LOCATE I%+2,18: PRINT USING "####.##";DF(I%+18*(PA
        GE%-1))
1490    NEXT
1500 FOR I% = 1 TO 18
1510    LOCATE I%+2,29: PRINT USING "####.##";YF(I%+18*(PA
        GE%-1))
1520    NEXT
1530 FOR I% = 1 TO 18
1540    LOCATE I%+2,40: PRINT USING "####.##";PM(I%+18*(PA
        GE%-1))
1550    NEXT
1560 FOR I% = 1 TO 18
1570    LOCATE I%+2,51: PRINT USING "####.##";RM(I%+18*(PA
        GE%-1))
1580    NEXT
1590 FOR I% = 1 TO 18
1600    LOCATE I%+2,62: PRINT USING "####.##";YM(I%+18*(PA
        GE%-1))
1610    NEXT
1620 FOR I% = 1 TO 18
1630    LOCATE I%+2,71: PRINT USING "####.##";AOA(I%+18*(P
        AGE%-1))
1640    NEXT
1650 LOCATE 22,7:PRINT USING "###";Q
1660 RETURN
1670 '----------ROW INDEXING---------------
1680 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
1690 OLDROW% = ROW%: ROW% = ROW% + FWD%
```

Figure A.12 ADATA.BAS (cont.)

```
1700 IF ROW% > NOD%+2 THEN ROW% = 3
1710 IF ROW% > 20 THEN ROW% = 3
1720 IF ROW% < 3 THEN ROW% = 3
1730 LOCATE OLDROW%,2: PRINT USING "###";OLDROW%-2+18*(PAG
     E%-1);: PRINT ":"
1740 COLOR 14,4: LOCATE ROW%,2
1750 PRINT USING "###";ROW%-2+18*(PAGE%-1);: PRINT ":"
1760 COLOR 15,1,7
1770 RETURN
1780 '----------COLUMN INDEXING----------------
1790 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
1800 OLDCOL% = COL%: COL% = COL% + FWD%
1810 IF COL% > 7 THEN COL% = 1
1820 IF COL% < 1 THEN COL% = 7
1830 IF OLDCOL% = 1 THEN LOCATE 1,9: PRINT "LIFT": GOTO
     1900 ELSE
1840 IF OLDCOL% = 2 THEN LOCATE 1,20: PRINT "DRAG": GOTO
     1900 ELSE
1850 IF OLDCOL% = 3 THEN LOCATE 1,32: PRINT "YAW": GOTO
     1900 ELSE
1860 IF OLDCOL% = 4 THEN LOCATE 1,41: PRINT "PITCH": GOTO
     1900 ELSE
1870 IF OLDCOL% = 5 THEN LOCATE 1,53: PRINT "ROLL": GOTO
     1900 ELSE
1880 IF OLDCOL% = 6 THEN LOCATE 1,62: PRINT "YAW M.": GOTO
     1900 ELSE
1890 IF OLDCOL% = 7 THEN LOCATE 1,74: PRINT "AOA": GOTO
     1900 ELSE
1900 COLOR 14,4
1910 IF COL% = 1 THEN LOCATE 1,9: PRINT "LIFT": GOTO 1980
     ELSE
1920 IF COL% = 2 THEN LOCATE 1,20: PRINT "DRAG": GOTO 1980
     ELSE
1930 IF COL% = 3 THEN LOCATE 1,32: PRINT "YAW": GOTO 1980
     ELSE
1940 IF COL% = 4 THEN LOCATE 1,41: PRINT "PITCH": GOTO 1980
     ELSE
1950 IF COL% = 5 THEN LOCATE 1,53: PRINT "ROLL": GOTO 1980
     ELSE
1960 IF COL% = 6 THEN LOCATE 1,62: PRINT "YAW M.": GOTO
     1980 ELSE
1970 IF COL% = 7 THEN LOCATE 1,74: PRINT "AOA": GOTO 1980
     ELSE
1980 COLOR 15,1,7
1990 RETURN
2000 '-------CHECK COMMANDS--------------------
2010 I% = 0
2020 WHILE I% < 7 AND VALID% = 0
2030    I% = I% + 1
2040    IF REP2$ = COMMAND$(I%) THEN FLAG% = I%: VALID% = 1
```

Figure A.12 ADATA.BAS (cont.)

```
2050    WEND
2060    IF VALID% = 0 THEN GOSUB 2420  'ERROR MESSAGE
2070 RETURN
2080 '----------SCROLL DOWN-----------------
2090 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
2100 PAGE% = PAGE% + FWD%
2110 IF PAGE% < 1 THEN PAGE% = 1
2120 GOSUB 1050  'PRINT BACKGROUND LINES
2130 ROW% = 3: COL% = 1: GOSUB 1740: GOSUB 1900 'HIGHLIGHT
     FIRST ROW AND COLUMN
2140 GOSUB 1430 'PRINT COLUMNS 1-7 AND Q VALUE
2150 OLDXPOS% = 0
2160 RETURN
2170 '---------SCROLL UP--------------------
2180 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
2190 PAGE% = PAGE% - FWD%
2200 IF PAGE% < 1 THEN PAGE% = 1
2210 GOSUB 1050 'PRINT BACKGROUND LINES
2220 ROW% = 3: COL% = 1: GOSUB 1740: GOSUB 1900 'HIGHLIGHT
     FIRST ROW AND COLUMN
2230 GOSUB 1430 ' PRINT COLUMNS 1-7 AND Q VALUE
2240 OLDXPOS% = 0
2250 RETURN
2260 '--------PRINT OUT Q VALUE--------------
2270 LOCATE 23,12: PRINT BLANK$
2280 LOCATE 23,12: INPUT "Q =";Q
2290 LOCATE 23,12: PRINT BLANK$
2300 COLOR 0,10
2310 LOCATE 22,7: PRINT USING "###";Q
2320 COLOR 15,1,7
2330 RETURN
2340 '----------EXIT EDITOR-----------------
2350 IF SAVED% = 1 OR MODIFIED% = 0 THEN GOTO 2410
2360 LOCATE 23,20: PRINT BLANK2$: BEEP
2370 LOCATE 23,20: INPUT "SAVE FILE (Y/N)";REP$
2380 IF REP$ = "N" OR REP$ = "n" THEN GOTO 2410
2390 GOSUB 2770   'SAVE FILE
2400 LOCATE 23,20: PRINT BLANK2$
2410 RETURN
2420 '-----------ERROR MESSAGE--------------
2430 LOCATE 23,20: PRINT BLANK2$
2440 LOCATE 23,20: PRINT " INVALID COMMAND"
2450 FOR F = 300 TO 500 STEP 100
2460    SOUND F,2
2470    SOUND 32767,2
2480    NEXT
2490 FOR I = 1 TO 500 STEP 1
2500 NEXT I
2510 LOCATE 23,20: PRINT BLANK2$
2520 RETURN
```

Figure A.12 ADATA.BAS (cont.)

```
2530 '---------ADD TEST POINTS TO FILE----------
2540 CLS
2550 LOCATE 3,5: PRINT"THIS OPTION ALLOWS YOU TO INCREASE
     THE NUMBER OF TEST"
2560 LOCATE 4,5: PRINT"POINTS OF AN EXISTING FILE"
2570 LOCATE 7,5: INPUT"WHAT IS THE NAME OF YOUR FILE";FILE$
2580 FILE$ = FD$ + FILE$
2590 GOSUB 2680
2610 LOCATE 10,5: INPUT"WHAT IS THE NEW NUMBER OF TUNNEL
     RUNS";NOQ%
2620 LOCATE 13,5: INPUT"WHAT IS THE NEW NUMBER OF TEST
     POINTS PER RUN";NOD%
2630 GOSUB 2770
2660 LOCATE 20,5: PRINT"NOW YOU CAN CALL UP YOUR FILE AND
     ADD IN THE NEW POINTS"
2670 RETURN
2680 '-------------READ IN FILE---------------
2690 OPEN FILE$ FOR INPUT AS #1
2700 INPUT #1,NOD%,CONFIG$,CONF
2710 INPUT #1,Q,W
2714 INPUT #1,ZL,ZD,ZY,ZPM,ZYM,ZRM,ZAOA
2715 INPUT #1,CLL,CALD,CALY,CALPM,CALYM,CALRM,CALAOA
2720   FOR J = 1 TO NOD%
2730   INPUT #1,LF(J),DF(J),YF(J),PM(J),RM(J),YM(J),AOA(J)
2740   NEXT J
2750 CLOSE #1
2760 RETURN
2770 '------------SAVE FILE-------------------
2780 LOCATE 23,20: PRINT BLANK2$: LOCATE 23,30: PRINT
     "SAVING FILE"
2790 OPEN FILE$ FOR OUTPUT AS #2
2800 WRITE #2,NOD%,CONFIG$,CONF
2810 WRITE #2,Q,W
2814 WRITE #2,ZL,ZD,ZY,ZPM,ZYM,ZRM,ZAOA
2815 WRITE #2,CLL,CALD,CALY,CALPM,CALYM,CALRM,CALAOA
2820  FOR J = 1 TO NOD%
2830    WRITE #2,LF(J),DF(J),YF(J),PM(J),RM(J),YM(J),AOA(J)
2840    NEXT J
2850 CLOSE #2
2860 RETURN
```

Figure A.12 ADATA.BAS (cont.)

```
10    REM----COMP.BAS;  CALCULATE Cd, Cl AND E. F. A.-----
15    REM
16    DIM LF(100),DF(100),YF(100),PM(100),YM(100),RM(100),
      AOA(100)
17    DIM CD(100),CL(100),EFA(100)
20    COLOR 15,1,4:KEY OFF: CLS
30    PRINT "WHAT IS THE NAME OF YOUR FILE CONTAINING THE";
35    PRINT " CALIBRATED"
40    INPUT "WIND TUNNEL DATA";FILE$
50    FILE$ = FD$+ FILE$
60    GOSUB 680         'READ IN DATA FROM FILE
70    REM -- FOR THE FOLLOWING, AREAL IS MEASURED IN SQUARE
      FEET
90    AREAL =60/144
270   FOR J = 1 TO NOD%
280       CL(J) = LF(J)/(Q*AREAL)
290       CD(J) = DF(J)/(Q*AREAL)
295       EFA(J) =LF(J)/Q
300       NEXT J
310   '-----------STORE DATA FOR Cd VS. Cl--------------
311   INPUT"WHAT IS THE NAME FOR THE FILE TO STORE Cd VS.
      Cl DATA";N$
312   Y$="Cd":X$="Cl"
320   OPEN FD$+N$ FOR OUTPUT AS #1
330   WRITE #1,NOD%,Y$,X$,CONFIG$,CONF
340   WRITE #1,Q
350       FOR J =  1 TO NOD%
360          WRITE #1,CD(J),CL(J)
370          NEXT J
380   CLOSE #1
390   '-----------STORE DATA FOR Cd VS. Cl*Cl------------
391   INPUT"WHAT IS THE NAME FOR THE FILE TO STORE Cd VS.
      Cl^2 DATA";N$
392   Y$="Cd":X$="Cl^2"
400   OPEN FD$+N$ FOR OUTPUT AS #2
410   WRITE #2,NOD%,Y$,X$,CONFIG$,CONF
420   WRITE #2,Q
430       FOR J =  1 TO NOD%
440          B = CL(J)*CL(J)
450          WRITE #2,CD(J),B
460          NEXT J
470   CLOSE #2
480   '-----------STORE DATA FOR Cl VS. AOA-------------
481   INPUT"WHAT IS THE NAME FOR THE FILE TO STORE THE Cl
      VS. AOA DATA";N$
482   Y$="Cl":X$="AOA"
490   OPEN FD$+N$ FOR OUTPUT AS #2
500   WRITE #2,NOD%,Y$,X$,CONFIG$,CONF
510   WRITE #2,Q
520       FOR J =  1 TO NOD%
```

Figure A.13 COMP.BAS - Program to Calculate Parameters

```
530           WRITE #2,CL(J),AOA(J)
540           NEXT J
550    CLOSE #2
560    '-----------STORE DATA FOR E.F.A. VS AOA-----------
561    INPUT"WHAT IS THE NAME FOR THE FILE TO STORE THE
        .F.A. VS. AOA DATA";N$
562    Y$="E.F.A.":X$="AOA"
570    OPEN FD$+N$ FOR OUTPUT AS #1
580    WRITE #1,NOD%,Y$,X$,CONFIG$,CONF
590    WRITE #1,Q
600        FOR J =  1 TO NOD%
610           WRITE #1,EFA(J),AOA(J)
620           NEXT J
630    CLOSE #1
640    '--------RETURN TO MAIN PROGRAM---------------------
650    COMMON FD$,PD$,DX%
660    CHAIN PD$+"MAIN.BAS",2110
670    '------------READ IN DATA FROM CALIBRATED FILE-----
680    OPEN FILE$ FOR INPUT AS #2
690    INPUT #2,NOD%,CONFIG$,CONF
700    INPUT #2,Q,W
701    INPUT #2,ZL,ZD,ZY,ZPM,ZYM,ZRM,ZAOA
702    INPUT #2,CLL,CALD,CALY,CALPM,CALYM,CALRM,CALAOA
710        FOR J = 1 TO NOD%
720        INPUT #2,LF(J),DF(J),YF(J),PM(J),RM(J),YM(J),
           AOA(J)
730        NEXT J
740    CLOSE #2
750    RETURN
```

Figure A.13 COMP.BAS (cont.)

```
10   '--------DATA EDITOR---------
20   GOSUB 180
30   COLOR 15,1,7: KEY OFF: CLS
40   LOCATE 5,34: PRINT "*OPTIONS MENU*"
50   PRINT
60   PRINT
70   PRINT TAB(30) "1.  CREATE A NEW FILE"
80   PRINT
90   PRINT TAB(30) "2.  EDIT EXISTING FILE"
100   PRINT
110  PRINT TAB(30) "3.  INCREASE NUMBER OF TEST POINTS"
120  PRINT TAB(30) "    IN AN EXISTING FILE"
130  PRINT
140  PRINT TAB(30) "4.  EXIT DATA EDITOR"
150  LOCATE 20,5: INPUT "ENTER YOUR CHOICE (1,2,3,4)";REP%
160  ON REP% GOSUB 380,270,2200,490
170  GOTO 30
180  '---------INITIALIZING CONSTANTS-------------
190  OPTION BASE 1: KEY OFF
200  DIM Y(100),X(100),COMMAND(10)
210  BLANK2$ = "                              "
220  BLANK1$ = "         "
230  BLANK$  = "                  "
240  COMMAND$(1) = "c":COMMAND$(2) = "r": COMMAND$(3) = "d"
250  COMMAND$(4) = "e": COMMAND$(5) = "s": COMMAND$(6) =
     "u": COMMAND$(7) = "q"
260  RETURN
270  '-----------EDIT A FILE----------------------
280  CLS: COLOR 15,1,7
290  LOCATE 5,10: INPUT "ENTER YOUR FILE NAME";FILE$
295  FILE$ = FD$ + FILE$
300  GOSUB 2350   ' READ IN FILE
310  PAGE% = 1: COL% = 1: SAVED% = 0: OLDXPOS% = 0:
     OLDYPOS% = 0
320  GOSUB 1010  'PRINT BACKGROUND LINES
330  ROW% = 3: GOSUB 1520 :  GOSUB 1620     'HIGHLIGHT
     FIRST ROW AND COLUMN
340  LOCATE 23,30: PRINT "LISTING": BEEP
350  GOSUB 1360  'PRINT COLUMNS 1-7 AND Q VALUE
360  LOCATE 23,20: PRINT BLANK2$
370  GOTO 480
380  '----------CREATE A FILE---------------------
390  COLOR 15,1,7: CLS
400  LOCATE 5,10: INPUT "ENTER YOUR FILE NAME";FILE$
405  FILR$ = FD$ + FILE$
410  LOCATE 7,10: INPUT "ENTER TEST CONFIGURATION";CONFIG$
420  LOCATE 9,10: INPUT "ENTER THE NUMBER OF TEST POINTS
     PER RUN";NOD%
430  LOCATE 11,10: INPUT "ENTER TITLE FOR X-VALUES (i.e.
     CL, AOA)";X1$
```

Figure A.14 BDATA.BAS - On-Screen Data Editor

```
440  LOCATE 13,10: INPUT "ENTER TITLE FOR Y-VALUES (i.e.
     Cd, Cl)";Y1$
450  PAGE% = 1: COL% = 1: SAVED% = 0: OLDXPOS% = 0:
     OLDYPOS% = 0
460  GOSUB 1010  'PRINT BACKGROUND LINES
470  ROW% = 3: GOSUB 1520: GOSUB 1620      'HIGHLIGHT FIRST
     ROW AND COLUMN
480  GOSUB 500  ' INSERT NEW DATA
490  COMMON FD$, PD$,DX%
491  CHAIN PD$+"MAIN.BAS",2110
492  RETURN
500  '----------CHANGE OR INSERT DATA--------------
510  LOCATE 23,12: PRINT BLANK$
520  LOCATE 23,12: INPUT REP$: IF REP$ = "" THEN GOTO 540
530  GOTO 580
540  LOCATE 23,12: PRINT BLANK$: LOCATE 23,12: PRINT
     "INVALID COMMAND": BEEP
550   FOR I = 1 TO 500 STEP 1
560   NEXT I
570  GOTO 510
580  REP2$ = RIGHT$(REP$,1)
590  T1% = ASC(REP2$)
600  IF T1% >= 43 AND T1% <= 57 THEN GOTO 660
610  T1% = T1% OR 32: REP2$ = CHR$(T1%)
620  GOSUB 1670
630  IF VALID% = 0 THEN GOTO 510 ELSE VALID% = 0
640  ON FLAG% GOSUB 1550,1450,1750,2010,2440,1840,1930
650  IF REP2$ <> "e" THEN GOTO 510 ELSE GOTO 1000
660  WHILE REP2$ <> "e"
670     NEWDATA = VAL(REP$)
680     IF QUIT% = 1 OR OLDXPOS% = 0 THEN QUIT% = 0: GOTO
     710   ELSE
690           LOCATE OLDXPOS%,OLDYPOS%,0
700           PRINT USING "####.##";OLDDATA
710     LOCATE 23,12: PRINT BLANK$
720     R% = ROW% - 2
730     IF COL% = 1 THEN Y(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 18: GOTO 750: ELS
740     IF COL% = 2 THEN X(R%+18*(PAGE%-1))=NEWDATA: YPOS%
        = 43: GOTO 750: ELSE
750     XPOS% = ROW%
760     LOCATE XPOS%,YPOS%: COLOR 0,10
770     PRINT USING "####.##"; NEWDATA: MODIFIED% = 1
780     SAVED% = 0: COLOR 15,1
790     OLDYPOS% = YPOS%: OLDXPOS% = XPOS%: OLDDATA =
        NEWDATA
800  FWD% = 1
810  I% = ROW%+1: GOSUB 1470  'HIGHLIGHT NEW ROW
820  IF I% > NOD%+2 THEN FWD% = 1 ELSE FWD% = 0
830  IF I% > 20 THEN FWD% = 1 ELSE FWD% = 0
```

Figure A.14 BDATA.BAS (cont.)

```
840  GOSUB 1570  'HIGHLIGHT NEW ROW
850  COLOR 15,1,7
860     LOCATE 23,12: PRINT BLANK$
870     LOCATE 23,12: INPUT REP$: IF REP$ = "" THEN GOTO
        890
880     GOTO 910
890     LOCATE 23,12: PRINT BLANK$: LOCATE 23,12: PRINT
        "INVALID COMMAND": BEEP
900     GOTO 860
910     REP2$ = RIGHT$(REP$,1)
920     T1% = ASC(REP2$)
930     IF T1% >= 43 AND T1% <= 57 THEN GOTO 660
940     T1% = T1% OR 32: REP2$ = CHR$(T1%)
950     GOSUB 1670  'CHECK COMMANDS
960     IF VALID% = 0 THEN GOTO 510 ELSE VALID% = 0
970     ON FLAG% GOSUB 1550,1450,1750,2010,2440,1840,1930
        'COMMAND SUBROUTINES
980     IF REP2$ <> "e" THEN GOTO 860 ELSE GOTO 990
990     WEND
1000  RETURN
1010 '-------------PRINT BACKGROUND LINES------------
1020  COLOR 15,1,7
1030  CLS
1040 LOCATE 2,1
1050 FOR I% = 1 TO 80 : PRINT CHR$(220);: NEXT
1060  LOCATE 1,18: PRINT "Y = ";Y1$
1070  LOCATE 1,43: PRINT "X = ";X1$
1080  LOCATE 1,68: PRINT "COMMANDS"
1090 LOCATE 21,1
1100 FOR I% = 1 TO 80 : PRINT CHR$(220);: NEXT
1110 FOR I% = 1 TO 22
1120      LOCATE I%,1: PRINT CHR$(222)
1130      LOCATE I%,80: PRINT CHR$(222)
1140 NEXT
1150 FOR I% = 1 TO 21
1160      LOCATE I%,28: PRINT CHR$(179)
1170      LOCATE I%,53: PRINT CHR$(179)
1180 NEXT
1190 FOR I% = 1 TO 18
1200     LOCATE I%+2,2
1210     PRINT USING "###";I%+18*(PAGE%-1);: PRINT ":"
1220     NEXT
1230 COLOR 15,4
1240 LOCATE 3,66: PRINT "C COLUMN      "
1250 LOCATE 4,66: PRINT "R ROW         "
1260 LOCATE 5,66: PRINT "D DOWN        "
1270 LOCATE 6,66: PRINT "E EXIT        "
1280 LOCATE 7,66: PRINT "S SAVE        "
1290 LOCATE 8,66: PRINT "U UP          "
1300 LOCATE 9,66: PRINT "Q TUNNEL SPEED"
```

Figure A.14 BDATA.BAS (cont.)

```
1310 COLOR 14,0: LOCATE 23,44: PRINT CONFIG$
1320 COLOR 15,1,7
1330 LOCATE 22,2: PRINT "[Q =": LOCATE 22,12: PRINT "]"
1340 LOCATE 23,3: PRINT "COMMAND :"
1350 RETURN
1360 '---------PRINT COLUMNS 1-7 AND Q VALUE---------
1370 FOR I% = 1 TO 18
1380    LOCATE I%+2,18: PRINT USING "####.##";Y(I%+18*(PAG
        E%-1))
1390    NEXT
1400 FOR I% = 1 TO 18
1410    LOCATE I%+2,43: PRINT USING "####.##";X(I%+18*(PAG
        E%-1))
1420    NEXT
1430 LOCATE 22,7:PRINT USING "###";Q
1440 RETURN
1450 '----------ROW INDEXING----------------
1460 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
1470 OLDROW% = ROW%: ROW% = ROW% + FWD%
1480 IF ROW% > NOD%+2 THEN ROW% = 3
1490 IF ROW% > 20 THEN ROW% = 3
1500 IF ROW% < 3 THEN ROW% = 3
1510 LOCATE OLDROW%,2: PRINT USING "###";OLDROW%-2+18*(PAG
     E%-1);: PRINT ":"
1520 COLOR 14,4: LOCATE ROW%,2: PRINT USING "###";ROW%-2+1
     8*(PAGE%-1);:PRINT ":"
1530 COLOR 15,1,7
1540 RETURN
1550 '----------COLUMN INDEXING----------------
1560 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
1570 OLDCOL% = COL%: COL% = COL% + FWD%
1580 IF COL% > 2 THEN COL% = 1
1590 IF COL% < 1 THEN COL% = 2
1600 IF OLDCOL% = 1 THEN LOCATE 1,18: PRINT "Y = ";Y1$:
     GOTO 1620 ELSE
1610 IF OLDCOL% = 2 THEN LOCATE 1,43: PRINT "X = ";X1$:
     GOTO 1620 ELSE
1620 COLOR 14,4
1630 IF COL% = 1 THEN LOCATE 1,18: PRINT "Y = ";Y1$: GOTO
     1650 ELSE
1640 IF COL% = 2 THEN LOCATE 1,43: PRINT "X = ";X1$: GOTO
     1650 ELSE
1650 COLOR 15,1,7
1660 RETURN
1670 '-------CHECK COMMANDS--------------------
1680 I% = 0
1690 WHILE I% < 7 AND VALID% = 0
1700    I% = I% + 1
1710    IF REP2$ = COMMAND$(I%) THEN FLAG% = I%: VALID% = 1
1720    WEND
```

Figure A.14 BDATA.BAS (cont.)

```
1730    IF VALID% = 0 THEN GOSUB 2090  'ERROR MESSAGE
1740 RETURN
1750 '----------SCROLL DOWN----------------
1760 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
1770 PAGE% = PAGE% + FWD%
1780 IF PAGE% < 1 THEN PAGE% = 1
1790 GOSUB 1010  'PRINT BACKGROUND LINES
1800 ROW% = 3: COL% = 1: GOSUB 1520: GOSUB 1620 'HIGHLIGHT
     FIRST ROW AND COLUMN
1810 GOSUB 1360 'PRINT COLUMNS 1-7 AND Q VALUE
1820 OLDXPOS% = 0
1830 RETURN
1840 '---------SCROLL UP--------------------
1850 FWD% = VAL(REP$): IF FWD% = 0 THEN FWD% = 1
1860 PAGE% = PAGE% - FWD%
1870 IF PAGE% < 1 THEN PAGE% = 1
1880 GOSUB 1010 'PRINT BACKGROUND LINES
1890 ROW% = 3: COL% = 1: GOSUB 1520: GOSUB 1620 'HIGHLIGHT
     FIRST ROW AND COLUMN
1900 GOSUB 1360 ' PRINT COLUMNS 1-7 AND Q VALUE
1910 OLDXPOS% = 0
1920 RETURN
1930 '--------PRINT OUT Q VALUE--------------
1940 LOCATE 23,12: PRINT BLANK$
1950 LOCATE 23,12: INPUT "Q =";Q
1960 LOCATE 23,12: PRINT BLANK$
1970 COLOR 0,10
1980 LOCATE 22,7: PRINT USING "###";Q
1990 COLOR 15,1,7
2000 RETURN
2010 '----------EXIT EDITOR-----------------
2020 IF SAVED% = 1 OR MODIFIED% = 0 THEN GOTO 2080
2030 LOCATE 23,20: PRINT BLANK2$: BEEP
2040 LOCATE 23,20: INPUT "SAVE FILE (Y/N)";REP$
2050 IF REP$ = "N" OR REP$ = "n" THEN GOTO 2080
2060 GOSUB 2440   'SAVE FILE
2070 LOCATE 23,20: PRINT BLANK2$
2080 RETURN
2090 '-----------ERROR MESSAGE--------------
2100 LOCATE 23,20: PRINT BLANK2$
2110 LOCATE 23,20: PRINT " INVALID COMMAND"
2120 FOR F = 300 TO 500 STEP 100
2130    SOUND F,2
2140    SOUND 32767,2
2150    NEXT
2160 FOR I = 1 TO 500 STEP 1
2170 NEXT I
2180 LOCATE 23,20: PRINT BLANK2$
2190 RETURN
2200 '---------ADD TEST POINTS TO FILE----------
```

Figure A.14 BDATA.BAS (cont.)

```
2210 CLS
2220 LOCATE 3,5: PRINT"THIS OPTION ALLOWS YOU TO INCREASE
     THE NUMBER OF TEST"
2230 LOCATE 4,5: PRINT"POINTS OF AN EXISTING FILE"
2240 LOCATE 7,5: INPUT"WHAT IS THE NAME OF YOUR FILE";FILE$
2250 FILE$ = FD$ + FILE$
2260 GOSUB 2350
2290 LOCATE 13,5: INPUT"WHAT IS THE NEW NUMBER OF TEST
     POINTS PER RUN";NOD%
2300 GOSUB 2440
2330 LOCATE 20,5: PRINT"NOW YOU CAN CALL UP YOUR FILE AND
     ADD IN THE NEW POINTS"
2340 RETURN
2350 '-------------READ IN FILE---------------
2360 OPEN FILE$ FOR INPUT AS #1
2370 INPUT #1,NOD%,Y1$,X1$,CONFIG$,CONF
2380 INPUT #1,Q
2390    FOR J = 1 TO NOD%
2400       INPUT #1,Y(J),X(J)
2410       NEXT J
2420 CLOSE #1
2430 RETURN
2440 '------------SAVE FILE--------------------
2450 LOCATE 23,20: PRINT BLANK2$: LOCATE 23,30: PRINT
     "SAVING FILE"
2460 OPEN FILE$ FOR OUTPUT AS #2
2470 WRITE #2,NOD%,Y1$,X1$,CONFIG$,CONF
2480 WRITE #2,Q
2490    FOR J = 1 TO NOD%
2500       WRITE #2,Y(J),X(J)
2510       NEXT J
2520 CLOSE #2
2530 RETURN
```

Figure A.14 BDATA.BAS (cont.)

```
10   '------------PLOTTING ROUTINE-----------------
20   DIM X(300),Y(300),B(300),C(300),D(300)
30   KEY OFF: SCREEN 2
40   HDPOS% = 44: B$ = "              "
60   GOSUB 260
70   OPT$ = CURVE$(1)
80   ERASE X,Y,FILESTK$,CURVE$,B,C,D
90   CHAIN PD$+"MAIN.BAS",2110,ALL
150  '------------READ IN FILE-----------------------
160  OPEN FILE$ FOR INPUT AS #2
170  INPUT #2,NOD%,Y1$,X1$,H$,CONF
180  INPUT #2,Q
190  FOR I% = 1 TO NOD%
200      INPUT #2,Y(I%),X(I%)
210      NEXT
220  CLOSE #2
230  RETURN
260  '-----------INITIALIZING PLOTTER-----------------
270  FOR L% = 1 TO NOF%
280      FILE$ = FILESTK$(L%)
290      GOSUB 160
300      IF L%=1 THEN XMIN=X(1): XMAX=XMIN: YMIN=Y(1):
     YMAX=YMIN
310      FOR I%=1 TO NOD%
320          IF X(I%)<XMIN THEN XMIN=X(I%) ELSE IF X(I%)>
     XMAX THEN XMAX=X(I%)
330          IF Y(I%)<YMIN THEN YMIN=Y(I%) ELSE IF Y(I%)>
     YMAX THEN YMAX=Y(I%)
340          NEXT I%
350      NEXT L%
351  CLS
352  PRINT "AFTER CURVES ARE PLOTTED YOU HAVE THE OPTION
     OF OBTAINING A HARD"
353  PRINT "COPY BY PRESSING CTRL-PRTSC.  THE CURVE WILL
     REMAIN ON THE SCREEN"
354  PRINT "UNTIL YOU PRESS THE F2 KEY."
355  PRINT
360  INPUT "HEADDING FOR PLOT =";HEAD$
370  T1%=LEN(HEAD$): IF T1%>60 THEN HEAD$=LEFT$(HEAD$,60)
380  INPUT "X TITLE FOR THE PLOT =";XTITLE$
390  T1%=LEN(XTITLE$): IF T1%>60 THEN XTITLE$=LEFT$(XTITLE
     $,60)
400  INPUT "Y TITLE FOR THE PLOT =";YTITLE$
410  T1%=LEN(YTITLE$): IF T1%>60 THEN YTITLE$=LEFT$(YTITLE
     $,60)
420  PRINT "MINIMUM X VALUE = [DEFAULT:";XMIN;"]";: INPUT
     T1$
430  IF T1$ <> "" THEN TEMP=VAL(T1$) ELSE TEMP=XMIN
440  IF TEMP<=XMIN THEN OK%=1 ELSE OK%=0
450  IF T1$ <> "" THEN TEMP3 =XMIN: XMIN = TEMP
```

Figure A.15 PLOTTER.BAS - Screen Graphics Program

```
460  PRINT "MAXIMUM X VALUE = [DEFAULT:";XMAX;"]";: INPUT
     T1$
470  IF T1$ <> "" THEN XMAX=VAL(T1$)
480  PRINT "MINIMUM Y VALUE = [DEFAULT:";YMIN;"]";: INPUT
     T1$
490  IF T1$ <> "" THEN YMIN=VAL(T1$)
500  PRINT "MAXIMUM Y VALUE = [DEFAULT:";YMAX;"]";: INPUT
     T1$
510  IF T1$ <> "" THEN YMAX=VAL(T1$)
520  RANGE=XMAX-XMIN
530  IF OK%=1 THEN PTR%=1: STKPTR%=1: PGSTK%(1)=1 ELSE
     GOSUB 580
540  CLS: SCREEN 2
550  GOSUB 840
560  RETURN
570  '----------SET PAGES----------------------------
580  TEMP2=(TEMP-TEMP3)/RANGE+1: PG%=INT(TEMP2)
590  STKPTR%=PG%+1
600  FOR I% = 1 TO STKPTR%
610      T2=TEMP-RANGE*(I%-1)
620      J%=1
630      WHILE X(J%)<T2
640          J%=J%+1
650          WEND
660      PGSTK%(PG%-I%+2)=J%
670  NEXT I%
680  PTR%=PGSTK%(STKPTR%)
690  RETURN
700  '--------PLOT WITHOUT GRID----------------------
710  LINE(ORIGINX%,ORIGINY%)-(ORIGINX%+7*INTERVALX%,TEMP4%
     ),1,B
720  X%=ORIGINX%
730  FOR I% = 1 TO 8
740      LINE (X%,162)-(X%,165)
750      X%=X%+INTERVALX%
760      NEXT
770  T1%=ORIGINX%-8: T2%=ORIGINY%
780  FOR I% = 1 TO 5
790      LINE (T1%,T2%)-(ORIGINX%,T2%)
800      T2%=T2%+INTERVALY%
810      NEXT
820  RETURN
830  '---------PLOTTING ROUTINES-------------------
840  ORIGINY%=12: ORIGINX%=72: INTERVALY%=30: INTERVALX%=80
850  TEMP4%=ORIGINY%+5*INTERVALY%
870  VIEW(0,0)-(639,163): CLS: VIEW
890  GOSUB 710
900  IF LEFT$(GRD$,1)="G" THEN GOSUB 2400
910  X%=ORIGINX%: Y%=ORIGINY%
920  FOR L%=1 TO NOF%
```

Figure A.15 PLOTTER.BAS (cont.)

```
930       IF NOF%=1 THEN GOTO 960
940       FILE$=FILESTK$(L%)
950       GOSUB 160
960       XINDEX=XMIN
970       GOSUB 1410
980       IF L%=1 THEN GOSUB 1120
990       IF L% <> NOF% THEN PTR%=PGSTK%(STKPTR%)
1000      NEXT L%
1001 ON KEY(1) GOSUB 1006
1002 ON KEY(2) GOSUB 1009
1003 KEY(2) OFF: KEY(1) ON: KEY(2) ON
1004 IF FLAG = 1 THEN GOTO 1100
1005 GOTO 1001
1006 LOCATE 23,1: PRINT "HI"
1008 RETURN
1009 FLAG = 1
1010 RETURN
1100 FLAG = 0
1101 RETURN
1110 '---------PLOT ZERO LINE-----------------
1120 IF (YMAX*YMIN <0) THEN TEMP=(-1*YMIN/(YMAX-YMIN))*150
     ELSE GOTO 1150
1130 DELTAY%=CINT(TEMP)
1140 LINE(ORIGINX%,TEMP4%-DELTAY%)-(ORIGINX%+7*INTERVALX%,
     TEMP4%-DELTAY%)
1150 IF (XMAX*XMIN <0) THEN TEMP=(-1*XMIN/(XMAX-XMIN))*550
     ELSE GOTO 1190
1160 DELTAX%=CINT(TEMP)
1170 LINE(ORIGINX%+DELTAX%,ORIGINY%)-(ORIGINX%+DELTAX%,TEM
     P4%)
1180 '---------PRINT INDECIES-------------------
1190 YINDEX=YMAX
1200 TEMP = (YMAX-YMIN)/5
1210 P%=LEN(HEAD$): P%=HDPOS%-(P%*.5)
1220 LOCATE 1,1: PRINT TAB(P%) HEAD$
1230 FOR I% = 1 TO 6
1240      PRINT USING "###.###";YINDEX
1250      IF (I%= 3) THEN FOR J% = 1 TO 2: PRINT: NEXT J%:
     GOTO 1260
1255      IF (I%<>6) THEN FOR J% = 1 TO 3: PRINT: NEXT J%
1260      YINDEX=YINDEX-TEMP
1270      NEXT I%
1280 XINDEX = XMIN: XINCR=(XMAX-XMIN)/7
1285 LOCATE 22,1
1290 PRINT USING "########.#";XINDEX;
1300 FOR I% = 1 TO 7
1310      XINDEX = XINDEX + XINCR
1320      PRINT USING "########.#";XINDEX; : NEXT I%
1330 P%=LEN(XTITLE$): P%=HDPOS%-(P%*.5)
1340 LOCATE 23,1: PRINT TAB(P%) XTITLE$
```

Figure A.15 PLOTTER.BAS (cont.)

```
1350 T1%=LEN(YTITLE$): P%=11-(T1%*.5)
1360 FOR I% = 1 TO T1%
1370     T1$=MID$(YTITLE$,I%,1): LOCATE P%+I%,1: PRINT T1$
1380     NEXT
1395 RETURN
1400 '-------PLOT POINTS ON GRAPH-------------
1410 TEMP2=1/(YMAX-YMIN): TEMP3=1/(XMAX-XMIN)
1420 IF MRK$="" THEN MRK$="MARK   "
1430 REP2$=LEFT$(MRK$,1)
1440 T%=L%: REP$=LEFT$(CURVE$(L%),1)
1450 IF T%>3 THEN T%=T%-3 : GOTO 1450
1460 IF REP$="C" THEN GOSUB 1870
1470 COUNT%=0: NDP%=0
1480 WHILE (PTR%<=NOD%) AND (X(PTR%) <= XMAX)
1490     IF X(PTR%) < XMIN THEN GOTO 1650
1500     OLDY%=Y%: OLDX%=X%
1510     NDP% = NDP%+1
1520     TEMP1=ABS((X(PTR%)-XMIN)*TEMP3)*550
1530     TEMP=ABS((Y(PTR%)-YMIN)*TEMP2)*150
1540     DELTAY%=CINT(TEMP): DELTAX%=CINT(TEMP1)
1560     Y%=TEMP4%-DELTAY% : X%=ORIGINX%+DELTAX%
1570     IF COUNT%=0 THEN OLDY%=Y%: OLDX%=X%: PSET(X%,Y%):
         FIRSTX%=X%: FIRSTY%=Y%: COUNT%=1
1580     IF REP$="P" THEN 1600
1590     LINE-(X%,Y%),1
1600     IF REP2$ <> "M" AND REP <> "P" THEN GOTO 1630
1610     IF T%=1 THEN CIRCLE(X%,Y%),2,1,0,6.28,5/12
1611     IF T%=2 THEN LINE(X%,Y%)-(X%+6,Y%): LINE-(X%+3,Y%
         -3): LINE-(X%,Y%)
1612     ELSE LINE(X%,Y%)-(X%+6,Y%-3),1,B : PSET(X%,Y%)
1620     GOTO 1640
1630     PSET(X%,Y%)
1640     IF REP$="C" THEN GOSUB 2140
1650     PTR%=PTR%+1
1660     WEND
1720 RETURN
1860 '--------CUBIC INTERPOLATION-------------
1870 DIM H(200),A(200),L(200),U(200),Z(200)
1880 FOR I% = 1 TO NOD%-1
1890     H(I%)=X(I%+1)-X(I%)
1900     NEXT
1910 FOR I%=2 TO NOD%-1
1920     T1=Y(I%+1)*H(I%-1)
1930     T2=Y(I%)*(X(I%+1)-X(I%-1))
1940     T3=Y(I%-1)*H(I%)
1950     A(I%)=3*(T1-T2+T3)/(H(I%-1)*H(I%))
1960     NEXT I%
1970 L(1)=1: U(1)=0: Z(1)=0
1980 FOR I%=2 TO NOD%
1990     L(I%)=2*(X(I%+1)-X(I%-1))-H(I%-1)*U(I%-1)
```

Figure A.15 PLOTTER.BAS (cont.)

```
2000      U(I%)=H(I%)/L(I%)
2010      Z(I%)=(A(I%)-H(I%-1)*Z(I%-1))/L(I%)
2020      NEXT I%
2030 L(NOD%)=1: Z(NOD%)=0: C(NOD%)=0
2040 FOR J%=NOD%-1 TO 1 STEP -1
2050      C(J%)=Z(J%)-U(J%)*C(J%+1)
2060      T1=(Y(J%+1)-Y(J%))/H(J%)
2070      T2=H(J%)*(C(J%+1)+2*C(J%))/3
2080      B(J%)=T1-T2
2090      D(J%)=(C(J%+1)-C(J%))/(3*H(J%))
2100      NEXT J%
2110 ERASE H,A,L,U,Z
2120 RETURN
2130 '---------PLOT CUBIC---------------------
2140 IF PTR%+1 > NOD% OR X(PTR%+1) > XMAX THEN GOTO 2280
2150 T1=(XMAX-XMIN)/560 : T2=X(PTR%)
2160 TEMP1=ABS((X(PTR%+1)-XMIN)*TEMP3)*550
2170 DELTAX%=CINT(TEMP1)
2180 XNEXT%=ORIGINX%+DELTAX%
2190 X%=X%+2
2200 IF X% >= XNEXT% THEN GOTO 2280
2210 T2=T2 + 2*T1
2220 T4=T2-X(PTR%)
2230 T3=Y(PTR%)+B(PTR%)*T4+C(PTR%)*T4*T4+D(PTR%)*T4*T4*T4
2240 TEMP=ABS((T3-YMIN)*TEMP2)*150
2250 DELTAY%=CINT(TEMP): Y%=TEMP4%-DELTAY%
2260 IF Y%>172 THEN Y%=172 ELSE IF Y%<12 THEN Y%=12
2270 LINE-(X%,Y%) : GOTO 2190
2280 RETURN
2400 '------PLOT WITH GRIDS----------------------
2410 FOR I% = 1 TO 9
2420      T1%=ORIGINY%+I%*15
2430      LINE(73,T1%)-(623,T1%),,,&H4444
2440      NEXT I%
2450 FOR I%= 1 TO 13
2460      T1%=ORIGINX%+I%*40
2470      LINE(T1%,12)-(T1%,162),,,&HAAAA
2480      NEXT I%
2490 RETURN
2500 END
```

Figure A.15 PLOTTER.BAS (cont.)

```
10 'NAME:  Data Acquisition And Control (DAAC)
20 '       HEADER for BASICA
30 '
40 'FILE NAME:  DACHDR.BAS
50 '
60 'DOS DEVICE NAME:  DAAC
70 '
80 'RESERVED FUNCTION NAMES:
90 '          AINM, AINS, AINSC, AOUM, AOUS,
100 '          BINM, BINS, BITINS, BITOUS, BOUM, BOUS,
110 '          CINM, CINS, CSET, DELAY
120 'RESERVED DEF SEG VALUE NAME:  DSEG
130 '
140 'NAMES DEFINED AND USED BY HEADER:
150 '          ADAPT%, AI, COUNT, FOUND%,
160 '          HNAME$, SG%, STAT%
170 '
180 '
190 'When using the BASICA Interpreter, this header
200 'must be executed before any function calls are
210 'made that access the DAAC adapter.  It initializes
220 'a number of variables for each function call.  These
230 'variables are reserved and should not be used except
240 'to access the DAAC adapter.  This routine also does a
250 'DEF SEG to the segment where the DAAC Device Driver
260 '(DAC.COM) is loaded.  If you execute a DEF SEG to
270 'access other hardware, you must DEF SEG to the segment
280 'of the DAAC Device Driver before any subsequent
290 'calls to access the DAAC adapter.
300 '
310 '
320 FOUND% = 0
330 SG% = &H2E
340 'Start searching the interrupt vectors until you find
350 'one that points to the DAAC device driver.
360 'Do a DEF SEG to that segment.
370 WHILE ((SG% <= &H3E) AND (FOUND% = 0))
380       DEF SEG = 0
390       DSEG = PEEK(SG%) + PEEK(SG% + 1) * 256
400       DEF SEG = DSEG
410       HNAME$=""
420       FOR AI=10 TO 17
430                 HNAME$ = HNAME$ + CHR$(PEEK(AI))
440       NEXT AI
450       IF HNAME$ = "DAAC    " AND PEEK(18) + PEEK(19) <>
          0 THEN FOUND% = 1
460       SG% = SG% + 4
470 WEND
480 IF FOUND% = 0 THEN PRINT "ERROR: DEVICE DRIVER DAC.COM
    NOT FOUND" : END
```

Figure A.16 BALCAL.BAS - Balance Calibration Program

```
490 'Now initialize all function name variables for calls
500 'to access the device driver.
510 AINM         = PEEK(&H13) * 256 + PEEK(&H12)
520 AINS         = PEEK(&H15) * 256 + PEEK(&H14)
530 AINSC        = PEEK(&H17) * 256 + PEEK(&H16)
540 AOUM         = PEEK(&H19) * 256 + PEEK(&H18)
550 AOUS         = PEEK(&H1B) * 256 + PEEK(&H1A)
560 BINM         = PEEK(&H1D) * 256 + PEEK(&H1C)
570 BINS         = PEEK(&H1F) * 256 + PEEK(&H1E)
580 BITINS       = PEEK(&H21) * 256 + PEEK(&H20)
590 BITOUS       = PEEK(&H23) * 256 + PEEK(&H22)
600 BOUM         = PEEK(&H25) * 256 + PEEK(&H24)
610 BOUS         = PEEK(&H27) * 256 + PEEK(&H26)
620 CINM         = PEEK(&H29) * 256 + PEEK(&H28)
630 CINS         = PEEK(&H2B) * 256 + PEEK(&H2A)
640 CSET         = PEEK(&H2D) * 256 + PEEK(&H2C)
650 DELAY        = PEEK(&H2F) * 256 + PEEK(&H2E)
660 'Finally, execute any call to re-initialize the
670 'device driver from any former invocation of BASIC.
680 ADAPT% = 0
690 COUNT = 1
700 STAT% = 0
710 CALL DELAY (ADAPT%, COUNT, STAT%)
720 '
730 'End of DAAC BASICA Header
740 '
750  REM------BALCAL.BAS  (CALIBRATE BALANCE)----
760  REM
765  DIM L(100),D(100),Y(100),PM(100),YM(100),RM(100),LBS(
     100)
769 DIM DAT(399),DAT%(399),DAT1(399),DAT1%(399)
770  FOR I = 1 TO 12
780     IF I = 1 THEN FILE$ = "LIFTP"
790     IF I = 2 THEN FILE$ = "DRAGP"
800     IF I = 3 THEN FILE$ = "YAWP"
810     IF I = 4 THEN FILE$ = "PITCHMP"
820     IF I = 5 THEN FILE$ = "YAWMP"
830     IF I = 6 THEN FILE$ = "ROLLMP"
840     IF I = 7 THEN FILE$ = "LIFTN"
850     IF I = 8 THEN FILE$ = "DRAGN"
860     IF I = 9 THEN FILE$ = "YAWN"
870     IF I = 10 THEN FILE$ = "PITCHMN"
880     IF I = 11 THEN FILE$ = "YAWMN"
890     IF I = 12 THEN FILE$ = "ROLLMN"
900  REM
910  REM
920  GOSUB 1380  'RECORD CALIBRATION DATA
930  NEXT I
940  CLS
950  LOCATE 10,5:PRINT "CALCULATING CALIBRATION CONSTANTS,
```

Figure A.16 BALCAL.BAS (cont.)

```
     PLEASE WAIT"
960  GOSUB 2100  'CALCULATE K1&K2 FOR ALL PRIME GAGES
970  GOSUB 2640  'CONVERT COUNTS TO FORCES
980  GOSUB 3180  'CALCULATE INTERACTION CONSTANTS
990  REM
1000  REM  STORE CALIBRATION CONSTANTS
1010  OPEN "C:CONST" FOR OUTPUT AS #1
1020  WRITE #1,INCALL,INCALD,INCALY,INCALPM,INCALYM,INCALRM
1040  WRITE #1,K1LPOS,K2LPOS,K1DPOS,K2DPOS,K1YPOS,K2YPOS
1050  WRITE #1,K1PMPOS,K2PMPOS,K1YMPOS,K2YMPOS,K1RMPOS,
      K2RMPOS
1060  WRITE #1,K1LNEG,K2LNEG,K1DNEG,K2DNEG,K1YNEG,K2YNEG
1070  WRITE #1,K1PMNEG,K2PMNEG,K1YMNEG,K2YMNEG,K1RMNEG,
      K2RMNEG
1080  WRITE #1,DDDL1P,DDDL2P,DYDL1P,DYDL2P,DPMDL1P,DPMDL2P
      ,DYMDL1P,DYMDL2P
1090  WRITE #1,DRMDL1P,DRMDL2P
1100  WRITE #1,DLDD1P,DLDD2P,DYDD1P,DYDD2P,DPMDD1P,DPMDD2P
      ,DYMDD1P,DYMDD2P
1110  WRITE #1,DRMDD1P,DRMDD2P
1120  WRITE #1,DLDY1P,DLDY2P,DDDY1P,DDDY2P,DPMDY1P,DPMDY2P
      ,DYMDY1P,DYMDY2P
1130  WRITE #1,DRMDY1P,DRMDY2P
1140  WRITE #1,DLDPM1P,DLDPM2P,DDDPM1P,DDDPM2P,DYDPM1P,DYD
      PM2P,DYMDPM1P,DYMDPM2P
1150  WRITE #1,DRMDPM1P,DRMDPM2P
1160  WRITE #1,DLDYM1P,DLDYM2P,DDDYM1P,DDDYM2P,DYDYM1P,DYD
      YM2P,DPMDYM1P,DPMDYM2P
1170  WRITE #1,DRMDYM1P,DRMDYM2P
1180  WRITE #1,DLDRM1P,DLDRM2P,DDDRM1P,DDDRM2P,DYDRM1P,DYD
      RM2P,DPMDRM1P,DPMDRM2P
1190  WRITE #1,DYMDRM1P,DYMDRM2P
1200  WRITE #1,DDDL1N,DDDL2N,DYDL1N,DYDL2N,DPMDL1N,DPMDL2N
      ,DYMDL1N,DYMDL2N
1210  WRITE #1,DRMDL1N,DRMDL2N
1220  WRITE #1,DLDD1N,DLDD2N,DYDD1N,DYDD2N,DPMDD1N,DPMDD2N
      ,DYMDD1N,DYMDD2N
1230  WRITE #1,DRMDD1N,DRMDD2N
1240  WRITE #1,DLDY1N,DLDY2N,DDDY1N,DDDY2N,DPMDY1N,DPMDY2N
      ,DYMDY1N,DYMDY2N
1250  WRITE #1,DRMDY1N,DRMDY2N
1260  WRITE #1,DLDPM1N,DLDPM2N,DDDPM1N,DDDPM2N,DYDPM1N,DYD
      PM2N,DYMDPM1N
1270  WRITE #1,DYMDPM2N,DRMDPM1N,DRMDPM2N
1280  WRITE #1,DLDYM1N,DLDYM2N,DDDYM1N,DDDYM2N,DYDYM1N,DYD
      YM2N,DPMDYM1N
1290  WRITE #1,DPMDYM2N,DRMDYM1N,DRMDYM2N
1300  WRITE #1,DLDRM1N,DLDRM2N,DDDRM1N,DDDRM2N,DYDRM1N,DYD
      RM2N,DPMDRM1N
1310  WRITE #1,DPMDRM2N,DYMDRM1N,DYMDRM2N
```

Figure A.16 BALCAL.BAS (cont.)

```
1320  CLOSE #1
1330  CLS: LOCATE 10,5
1340  PRINT " CALIBRATION COMPLETE"
1350  END
1360  REM
1370  REM
1380 REM---RECORD CALIBRATION DATA------
1401 COLOR 15,1: KEY OFF: CLS
1402 PRINT "CALIBRATION FOR THE LOADING OF THE ";FILE$;"
     COMPONENT"
1403 PRINT: INPUT "AFTER ALL AMPLIFIERS ARE ZEROED PRESS
     RETURN";X
1404 CLS
1405 PRINT "  ZD        ZL       ZY         ZPM       ZYM
     ZRM "
1406 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
1407 ADAPT%= 0: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=3
1408 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT%(0),STAT%)
1409 ZD = 0:ZPM=0:ZL=0:ZYM=0
1410 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
1411 FOR J = 0 TO 396 STEP 4
1412 DAT(J)=(DAT%(J)/204.8)-10
1413 ZD =ZD + DAT(J)
1414 NEXT J
1415 ZD = ZD/100
1416 FOR J = 1 TO 397 STEP 4
1417 DAT(J)=(DAT%(J)/204.8)-10
1418 ZL = ZL + DAT(J)
1419 NEXT J
1420 FOR J = 2 TO 398 STEP 4
1421 DAT(J)=(DAT%(J)/204.8)-10
1422 ZPM = ZPM + DAT(J)
1423 NEXT J
1424 FOR J = 3 TO 399 STEP 4
1425 DAT(J)=(DAT%(J)/204.8)-10
1426 ZYM = ZYM + DAT(J)
1427 NEXT J
1428 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
1429 ADAPT%= 1: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=1
1430 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT1%(0),STAT%)
1431 ZY =0:ZRM=0
1432 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
1435 FOR J = 0 TO 198 STEP 2
1436 DAT1(J)=(DAT1%(J)/204.8)-10
1437 ZRM = ZRM + DAT1(J)
1438 NEXT J
```

Figure A.16 BALCAL.BAS (cont.)

```
1439 ZRM = ZRM/100
1440 FOR J = 1 TO 199 STEP 2
1441 DAT1(J)=(DAT1%(J)/204.8)-10
1442 ZY = ZY + DAT1(J)
1443 NEXT J
1444 ZL=ZL/100:ZPM=ZPM/100:ZYM=ZYM/100:ZY=ZY/100
1445 LOCATE 3,1: PRINT USING "+#.###";ZD: LOCATE 3,10: PRI
     NT USING "+#.###";ZL
1446 LOCATE 3,19: PRINT USING "+#.###";ZY
1447 LOCATE 3,28: PRINT USING "+#.###";ZPM
1448 LOCATE 3,37: PRINT USING "+#.###";ZYM
1449 LOCATE 3,46: PRINT USING "+#.###";ZRM
1451 REM
1452 PRINT:PRINT
1453 INPUT "AFTER PLACING ALL CAL SWITCHES TO + SETTING HIT
     RETURN";X
1454 PRINT
1455 PRINT " CALD      CLL     CALY      CALPM      CALYM
     CALRM"
1456 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
1457 ADAPT%= 0: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=3
1458 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT%(0),STAT%)
1459 CALD=0:CLL=0:CALYM=0:CALPM=0
1460 IF STAT%<> 0,THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
1461 FOR J = 0 TO 396 STEP 4
1462 DAT(J)=(DAT%(J)/204.8)-10
1463 CALD =CALD + DAT(J)
1464 NEXT J
1465 CALD = CALD/100
1466 FOR J = 1 TO 397 STEP 4
1467 DAT(J)=(DAT%(J)/204.8)-10
1468 CLL = CLL + DAT(J)
1469 NEXT J
1470 FOR J = 2 TO 398 STEP 4
1471 DAT(J)=(DAT%(J)/204.8)-10
1472 CALPM = CALPM + DAT(J)
1473 NEXT J
1474 FOR J = 3 TO 399 STEP 4
1475 DAT(J)=(DAT%(J)/204.8)-10
1476 CALYM = CALYM + DAT(J)
1477 NEXT J
1478 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
1479 ADAPT%= 1: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=1
1480 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT1%(0),STAT%)
1481 CALY =0:CALRM=0
1482 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
```

Figure A.16 BALCAL.BAS (cont.)

```
1485 FOR J = 0 TO 198 STEP 2
1486 DAT1(J)=(DAT1%(J)/204.8)-10
1487 CALRM = CALRM + DAT1(J)
1488 NEXT J
1489 CALRM = CALRM/100
1490 FOR J = 1 TO 199 STEP 2
1491 DAT1(J)=(DAT1%(J)/204.8)-10
1492 CALY = CALY + DAT1(J)
1493 NEXT J
1494 CLL=CLL/100:CALPM=CALPM/100:CALYM=CALYM/100:CALY=CALY
     /100
1495 LOCATE 10,1: PRINT USING "+#.###";CALD
1496 LOCATE 10,10: PRINT USING "+#.###";CLL
1497 LOCATE 10,19: PRINT USING "+#.###";CALY
1498 LOCATE 10,28: PRINT USING "+#.###";CALPM
1499 LOCATE 10,37: PRINT USING "+#.###";CALYM
1500 LOCATE 10,46: PRINT USING "+#.###";CALRM
1501 REM
1502 REM
1503    IF I = 1 THEN INCALL = CLL - ZL
1510    IF I = 2 THEN INCALD = CALD - ZD
1520    IF I = 3 THEN INCALY = CALT - ZY
1530    IF I = 4 THEN INCALPM = CALPM - ZPM
1540    IF I = 5 THEN INCALYM = CALYM - ZYM
1550    IF I = 6 THEN INCALRM = CALRM - ZRM
1611 PRINT:INPUT "PRESS ENTER (RETURN) TO CONTINUE";X
1620 CLS:PRINT "RETURN ALL CAL SWITCHES TO CENTER POSITION"
1630 PRINT:
1640 PRINT "LOAD THE ";FILE$;" GAGE FROM 0 TO 20 POUNDS AND
     THEN BACK TO 0"
1650 PRINT "POUNDS IN 1 POUND INCREMENTS.  THERE SHOULD BE
     TWO READINGS FOR"
1660 PRINT "EACH POUND WEIGHT EXCEPT FOR THE LOAD AT 20
     POUNDS WHICH WILL HAVE"
1670 PRINT "ONLY ONE READING"
1680 PRINT
1690 PRINT "PRESS THE F1 KEY WHEN THE LOADING IS FINISHED"
1700 PRINT "PRESS THE F2 KEY WHEN READY TO RECORD THE DATA
     FOR THAT LOAD"
1701 PRINT: INPUT "PRESS ENTER (RETURN) TO CONTINUE";X
1702 CLS
1703 PRINT " DRAG     LIFT     YAW      PITCH     YAW
     ROLL  "
1704 PRINT "                                MOM.     MOM.
     MOM. "
1710 SOAP = 0: N=4
1720 FOR K = 1 TO 100
1730 ON KEY(1) GOSUB 1830     'SET STOP FLAG
1740 ON KEY(2) GOSUB 1841     'RECORD DATA
1750 KEY(1) ON: KEY(2) ON
```

Figure A.16 BALCAL.BAS (cont.)

```
1760 IF SOAP = 2 THEN GOTO 1780
1770 IF SOAP = 1 THEN GOTO 1920
1775 GOTO 1730
1780 SOAP = 0
1790 NOD% = K
1800 NEXT K
1810 GOTO 1920
1830 REM  SET STOP FLAG
1838 SOAP = 1
1839 RETURN
1840 REM STEPS TO RECORD DATA
1841 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
1842 ADAPT%= 0: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=3
1843 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT%(0),STAT%)
1844 D(K)=0:L(K)=0:YM(K)=0:PM(K)=0
1845 IF STAT%<> 0,THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
1846 FOR J = 0 TO 396 STEP 4
1847 DAT(J)=(DAT%(J)/204.8)-10
1848 D(K) =D(K) + DAT(J)
1849 NEXT J
1850 D(K) = D(K)/100
1851 FOR J = 1 TO 397 STEP 4
1852 DAT(J)=(DAT%(J)/204.8)-10
1853 L(K) = L(K) + DAT(J)
1854 NEXT J
1855 FOR J = 2 TO 398 STEP 4
1856 DAT(J)=(DAT%(J)/204.8)-10
1857 PM(K) = PM(K) + DAT(J)
1858 NEXT J
1859 FOR J = 3 TO 399 STEP 4
1860 DAT(J)=(DAT%(J)/204.8)-10
1861 YM(K) = YM(K) + DAT(J)
1862 NEXT J
1863 STAT%=0:  MODE%=0: STOR%=0: COUNT=100: RATE=500
1864 ADAPT%= 1: DEVICE%= 9: CHANLO%= 0: CTRL%= 0: CHANHI%=1
1865 CALL AINSC(ADAPT%,DEVICE%,CHANLO%,CHANHI%,CTRL%,MODE%
     ,STOR%,COUNT,RATE,DAT1%(0),STAT%)
1866 Y(K) =0:RM(K)=0
1867 IF STAT%<> 0 THEN PRINT USING "EXECUTION ERROR ###";
     STAT%:END
1868 FOR J = 0 TO 198 STEP 2
1869 DAT1(J)=(DAT1%(J)/204.8)-10
1870 RM(K) = RM(K) + DAT1(J)
1871 NEXT J
1872 RM(K) = RM(K)/100
1873 FOR J = 1 TO 199 STEP 2
1874 DAT1(J)=(DAT1%(J)/204.8)-10
1875 Y(K) = Y(K) + DAT1(J)
```

Figure A.16 BALCAL.BAS (cont.)

```
1876 NEXT J
1877 L(K)=L(K)/100:PM(K)=PM(K)/100:YM(K)=YM(K)/100:Y(K)=Y(
     K)/100
1878  IF N> 23 THEN N=4:CLS:PRINT " DRAG      LIFT       YAW
      PITCH      YAW      ROLL ":PRINT "
        MOM.       MOM.     MOM. "
1879 LOCATE N,1: PRINT USING "+#.###";D(K)
1880 LOCATE N,10: PRINT USING "+#.###";L(K)
1881 LOCATE N,19: PRINT USING "+#.###";Y(K)
1882 LOCATE N,28: PRINT USING "+#.###";PM(K)
1883 LOCATE N,37: PRINT USING "+#.###";YM(K)
1884 LOCATE N,46: PRINT USING "+#.###";RM(K)
1885 N=N+1
1900 SOAP = 2
1910 RETURN   'GO BACK FOR NEXT DATA POINT
1920 '---CORRECT FOR DRIFT------------
1921 DIFFL =(L(NOD%)-L(1)): DIFFD=(D(NOD%)-D(1)):DIFFY=(Y(
     NOD%)-Y(1))
1922 DIFFPM=(PM(NOD%)-PM(1)):DIFFYM=(YM(NOD%)-YM(1)):DIFFR
     M=(RM(NOD%)-RM(1))
1923 D=NOD%-1
1924 CORL=DIFFL/D:CORD=DIFFD/D:CORY=DIFFY/D:CORPM=DIFFPM/D
     :CORYM=DIFFYM/D:CORRM=DIFFRM/D
1925 A=1
1926 FOR K = 2 TO NOD%
1927   L(K)=L(K)-(A*CORL)
1928   D(K)=D(K)-(A*CORD)
1929   Y(K)=Y(K)-(A*CORY)
1930   PM(K)=PM(K)-(A*CORPM)
1931   YM(K)=YM(K)-(A*CORYM)
1932   RM(K)=RM(K)-(A*CORRM)
1933   A = A+1
1934 NEXT K
1935 '---WRITE DATA TO FILE-----
1940 FILE$ = "C:"+FILE$
1950 OPEN FILE$ FOR OUTPUT AS #1
1960 WRITE #1,ZL,ZD,ZY,ZPM,ZYM,ZRM,NOD%
1970 WRITE #1,CLL,CALD,CALY,CALPM,CALYM,CALRM
1971 IF I=1 OR I=2 OR I=3 THEN GOTO 2079
1972 IF I=7 OR I=8 OR I=9 THEN GOTO 2067
1976 LBS = 0
1977 FOR J = 1 TO 11
1978    IF I=4 OR I=10 THEN D1=20
1979    IF I=5 OR I=11 THEN D1=4.5
1980    IF I=6 OR I=12 THEN D1=11.5
1981    MOM = LBS*D1 'CONVERT TO MOMENTS
1982    IF I=10 OR I=11 OR I=12 THEN MOM=MOM*(-1)
1983    WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),MOM
1984    LBS = LBS+1
1985    NEXT J
```

Figure A.16 BALCAL.BAS (cont.)

```
1986 LBS = 9
1987 FOR J = 12 TO NOD%
1988    IF I=4 OR I=10 THEN D1=20
1989    IF I=5 OR I=11 THEN D1=4.5
1990    IF I=6 OR I=12 THEN D1=11.5
1991    MOM = LBS*D1 'CONVERT TO MOMENTS
1992    IF I=10 OR I=11 OR I=12 THEN MOM=MOM*(-1)
1993    WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),MOM
1994    LBS = LBS-1
1995    NEXT J
1996 CLOSE #1
1997 RETURN    'GO BACK TO SET UP NEXT CALIBRATION
2067 LBS = 0
2068 FOR J = 1 TO 21
2069    WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS
2070    LBS = LBS - 1
2071    NEXT J
2072 LBS = -19
2073 FOR J = 22 TO NOD%
2074    WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS
2075    LBS = LBS + 1
2076    NEXT J
2077 CLOSE #1
2078 RETURN    'GO BACK TO SET UP NEXT CALIBRATION
2079 LBS = 0
2080 FOR J = 1 TO 21
2081    WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS
2082    LBS = LBS + 1
2083    NEXT J
2084 LBS = 19
2085 FOR J = 22 TO NOD%
2086    WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS
2087    LBS = LBS - 1
2088    NEXT J
2089 CLOSE #1
2090 RETURN    'GO BACK TO SET UP NEXT CALIBRATION
2100 REM-----------------------------------------
2110 REM CALCULATE K1&K2 FOR PRIME GAGES
2120 FOR I = 1 TO 12
2130    IF I = 1 THEN FILE$ = "LIFTP"
2140    IF I = 2 THEN FILE$ = "DRAGP"
2150    IF I = 3 THEN FILE$ = "YAWP"
2160    IF I = 4 THEN FILE$ = "PITCHMP"
2170    IF I = 5 THEN FILE$ = "YAWMP"
2180    IF I = 6 THEN FILE$ = "ROLLMP"
2190    IF I = 7 THEN FILE$ = "LIFTN"
2200    IF I = 8 THEN FILE$ = "DRAGN"
2210    IF I = 9 THEN FILE$ = "YAWN"
2220    IF I = 10 THEN FILE$ = "PITCHMN"
2230    IF I = 11 THEN FILE$ = "YAWMN"
```

Figure A.16 BALCAL.BAS (cont.)

```
2240    IF I = 12 THEN FILE$ = "ROLLMN"
2250 GOSUB 2295   'READ FILES AND PERFORM CALCULATIONS
2260 NEXT I
2270 RETURN     'GO BACK TO CONVERT COUNTS
2280 REM
2290 REM
2295 A=0:B=0:C=0:D=0:E=0:
2300 OPEN "C:"+FILE$ FOR INPUT AS #2
2310 INPUT #2,ZL,ZD,ZY,ZPM,ZYM,ZRM,NOD%
2320 INPUT #2,CLL,CALD,CALY,CALPM,CALYM,CALRM
2340 FOR J = 1 TO NOD%
2350    INPUT #2,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS(J)
2360        IF I = 1 OR I = 7 THEN X = L(J)
2370        IF I = 2 OR I = 8 THEN X = D(J)
2380        IF I = 3 OR I = 9 THEN X = Y(J)
2390        IF I = 4 OR I = 10 THEN X = PM(J)
2400        IF I = 5 OR I = 11 THEN X = YM(J)
2410        IF I = 6 OR I = 12 THEN X = RM(J)
2420    A = A + (X^2)
2430    B = B + (X^3)
2440    C = C + (X^4)
2450    D = D + (X*LBS(J))
2460    E = E + (X*X*LBS(J))
2470    NEXT J
2480 CLOSE #2
2490 K2 = ((D/A)-(E/B))/((B/A)-(C/B))
2500 K1 = (D/A) - (K2*(B/A))
2510 IF I = 1 THEN K1LPOS = K1: K2LPOS = K2
2520 IF I = 2 THEN K1DPOS = K1: K2DPOS = K2
2530 IF I = 3 THEN K1YPOS = K1: K2YPOS = K2
2540 IF I = 4 THEN K1PMPOS = K1: K2PMPOS = K2
2550 IF I = 5 THEN K1YMPOS = K1: K2YMPOS = K2
2560 IF I = 6 THEN K1RMPOS = K1: K2RMPOS = K2
2570 IF I = 7 THEN K1LNEG = K1: K2LNEG = K2
2580 IF I = 8 THEN K1DNEG = K1: K2DNEG = K2
2590 IF I = 9 THEN K1YNEG = K1: K2YNEG = K2
2600 IF I = 10 THEN K1PMNEG = K1: K2PMNEG = K2
2610 IF I = 11 THEN K1YMNEG = K1: K2YMNEG = K2
2620 IF I = 12 THEN K1RMNEG = K1: K2RMNEG = K2
2630 RETURN   'GO BACK TO CALCULATE K1&K2 FOR NEXT FILE
2640 REM-----------------------------------------
2650 REM CONVERT COUNTS TO FORCES
2660 FOR I = 1 TO 12
2670    IF I = 1 THEN FILE$ = "LIFTP"
2680    IF I = 2 THEN FILE$ = "DRAGP"
2690    IF I = 3 THEN FILE$ = "YAWP"
2700    IF I = 4 THEN FILE$ = "PITCHMP"
2710    IF I = 5 THEN FILE$ = "YAWMP"
2720    IF I = 6 THEN FILE$ = "ROLLMP"
2730    IF I = 7 THEN FILE$ = "LIFTN"
```

Figure A.16 BALCAL.BAS (cont.)

```
2740    IF I = 8 THEN FILE$ = "DRAGN"
2750    IF I = 9 THEN FILE$ = "YAWN"
2760    IF I = 10 THEN FILE$ = "PITCHMN"
2770    IF I = 11 THEN FILE$ = "YAWMN"
2780    IF I = 12 THEN FILE$ = "ROLLMN"
2790 GOSUB 2835   'READ IN FILE AND CONVERT
2800 NEXT I
2810 RETURN    'GO BACK TO CALCULATE INTERACTON CONSTANTS
2820 REM
2830 REM
2835 A=0:B=0:C=0:D=0:E=0:F=0
2840 OPEN "C:"+FILE$ FOR INPUT AS #2
2850 INPUT #2,ZL,ZD,ZY,ZPM,ZYM,ZRM,NOD%
2860 INPUT #2,CLL,CALD,CALY,CALPM,CALYM,CALRM
2870 FOR J = 1 TO NOD%
2880    INPUT #2,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS(J)
2890    A=((INCALL/CLL-ZL)*(L(J)-ZL))
2900    B=((INCALD/CALD-ZD)*(D(J)-ZD))
2910    C=((INCALY/CALY-ZY)*(Y(J)-ZY))
2920    D=((INCALPM/CALPM-ZPM)*(PM(J)-ZPM))
2930    E=((INCALYM/CALYM-ZYM)*(YM(J)-ZYM))
2940    F=((INCALRM/CALRM-ZRM)*(RM(J)-ZRM))
2950    IF L(J) < 0 THEN K1 = K1LNEG: K2 = K2LNEG ELSE K1
        = K1LPOS:K2 = K2LPOS
2960    L(J)=(K1*A) + (K2*(A^2))
2970    IF D(J) < 0 THEN K1 = K1DNEG: K2 = K2DNEG ELSE K1
        = K1DPOS:K2 = K2DPOS
2980    D(J)=(K1*B) + (K2*(B^2))
2990    IF Y(J) < 0 THEN K1 = K1YNEG: K2 = K2YNEG ELSE K1
        = K1YPOS:K2 = K2YPOS
3000    Y(J)=(K1*C) + (K2*(C^2))
3010    IF PM(J) < 0 THEN K1=K1PMNEG:K2=K2PMNEG ELSE K1=K1
        PMPOS:K2=K2PMPOS
3020    PM(J)=(K1*D) + (K2*(D^2))
3030    IF YM(J) < 0 THEN K1=K1YMNEG:K2=K2YMNEG ELSE K1=K1
        YMPOS:K2=K2YMPOS
3040    YM(J)=(K1*E) + (K2*(E^2))
3050    IF RM(J) < 0 THEN K1=K1RMNEG:K2=K2RMNEG ELSE K1=K1
        RMPOS:K2=K2RMPOS
3060    RM(J)=(K1*F) + (K2*(F^2))
3070    NEXT J
3080 CLOSE #2
3090 REM  SAVE CONVERTED COUNTS
3100 OPEN "B:"+FILE$ FOR OUTPUT AS #1
3110 WRITE #1,ZL,ZD,ZY,ZPM,ZYM,ZRM,NOD%
3120 WRITE #1,CLL,CALD,CALY,CALPM,CALYM,CALRM
3130 FOR J = 1 TO NOD%
3140    WRITE #1,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS(J)
3150    NEXT J
3155 CLOSE #1
```

Figure A.16 BALCAL.BAS (cont.)

```
3160 RETURN   'GO BACK TO CALCULATE INTERACTION CONSTANTS
3170 REM-----------------------------------------------
3180 REM CALCULATE INTERACTION CONSTANTS
3190 FOR I = 1 TO 12
3200    IF I = 1 THEN FILE$ = "LIFTP"
3210    IF I = 2 THEN FILE$ = "DRAGP"
3220    IF I = 3 THEN FILE$ = "YAWP"
3230    IF I = 4 THEN FILE$ = "PITCHMP"
3240    IF I = 5 THEN FILE$ = "YAWMP"
3250    IF I = 6 THEN FILE$ = "ROLLMP"
3260    IF I = 7 THEN FILE$ = "LIFTN"
3270    IF I = 8 THEN FILE$ = "DRAGN"
3280    IF I = 9 THEN FILE$ = "YAWN"
3290    IF I = 10 THEN FILE$ = "PITCHMN"
3300    IF I = 11 THEN FILE$ = "YAWMN"
3310    IF I = 12 THEN FILE$ = "ROLLMN"
3320 GOSUB 3365  'READ IN FILES AND CALCULATE CONSTANTS
3330 NEXT I
3340 RETURN     'GO BACK TO SAVE CALIBRATION DATA
3350 REM
3360 REM
3365 A=0:B=0:C=0:D1=0:D2=0:D3=0:D4=0:D5=0:E1=0:E2=0:E3=0:E
     4=0:E5=0
3370 OPEN "B:"+FILE$ FOR INPUT AS #2
3380 INPUT #2,ZL,ZD,ZY,ZPM,ZYM,ZRM,NOD%
3390 INPUT #2,CLL,CALD,CALY,CALPM,CALYM,CALRM
3410 FOR J = 1 TO NOD%
3420    INPUT #2,L(J),D(J),Y(J),PM(J),YM(J),RM(J),LBS(J)
3430    IF I=1 OR I=7 THEN X=L(J):Y1=D(J):Y2=Y(J):Y3=PM(J)
     :Y4=YM(J):Y5=RM(J)
3440    IF I=2 OR I=8 THEN X=D(J):Y1=L(J):Y2=Y(J):Y3=PM(J)
     :Y4=YM(J):Y5=RM(J)
3450    IF I=3 OR I=9 THEN X=Y(J):Y1=L(J):Y2=D(J):Y3=PM(J)
     :Y4=YM(J):Y5=RM(J)
3460    IF I=4 OR I=10 THEN X=PM(J):Y1=L(J):Y2=D(J):Y3=Y(J)
     :Y4=YM(J):Y5=RM(J)
3470    IF I=5 OR I=11 THEN X=YM(J):Y1=L(J):Y2=D(J):Y3=Y(J)
     :Y4=PM(J):Y5=RM(J)
3480    IF I=6 OR I=12 THEN X=RM(J):Y1=L(J):Y2=D(J):Y3=Y(J)
     :Y4=PM(J):Y5=YM(J)
3490    A = A + (X^2)
3500    B = B + (X^3)
3510    C = C + (X^4)
3520    D1 = D1 + (X*Y1)
3530    E1 = E1 + ((X^2)*Y1)
3540    D2 = D2 + (X*Y2)
3550    E2 = E2 + ((X^2)*Y2)
3560    D3 = D3 + (X*Y3)
3570    E3 = E3 + ((X^2)*Y3)
3580    D4 = D4 + (X*Y4)
```

Figure A.16 BALCAL.BAS (cont.)

```
3590     E4 = E4 + ((X^2)*Y4)
3600     D5 = D5 + (X*Y5)
3610     E5 = E5 + ((X^2)*Y5)
3620     NEXT J
3630 CLOSE #2
3640 K12 = ((D1/A)-(E1/B))/((B/A)-(C/B))
3650 K11 = (D1/A)-(K12*(B/A))
3660 K22 = ((D2/A)-(E2/B))/((B/A)-(C/B))
3670 K21 = (D2/A)-(K22*(B/A))
3680 K32 = ((D3/A)-(E3/B))/((B/A)-(C/B))
3690 K31 = (D3/A)-(K32*(B/A))
3700 K42 = ((D4/A)-(E4/B))/((B/A)-(C/B))
3710 K41 = (D4/A)-(K42*(B/A))
3720 K52 = ((D5/A)-(E5/B))/((B/A)-(C/B))
3730 K51 = (D5/A)-(K52*(B/A))
3740 IF I=1 THEN DDDL1P=K11:DDDL2P=K12:DYDL1P=K21:DYDL2P=K
     22:DPMDL1P=K31
3750 IF I=1 THEN DPMDL2P=K32:DYMDL1P=K41:DYMDL2P=K42:DRMDL
     1P=K51:DRMDL2P=K52
3760 IF I=2 THEN DLDD1P=K11:DLDD2P=K12:DYDD1P=K21:DYDD2P=K
     22:DPMDD1P=K31
3770 IF I=2 THEN DPMDD2P=K32:DYMDD1P=K41:DYMDD2P=K42:DRMDD
     1P=K51:DRMDD2P=K52
3780 IF I=3 THEN DLDY1P=K11:DLDY2P=K12:DDDY1P=K21:DDDY2P=K
     22:DPMDY1P=K31
3790 IF I=3 THEN DPMDY2P=K32:DYMDY1P=K41:DYMDY2P=K42:DRMDY
     1P=K51:DRMDY2P=K52
3800 IF I=4 THEN DLDPM1P=K11:DLDPM2P=K12:DDDPM1P=K21:DDDPM
     2P=K22:DYDPM1P=K31
3810 IF I=4 THEN DYDPM2P=K32:DYMDPM1P=K41:DYMDPM2P=K42:DRM
     DPM1P=K51:DRMDPM2P=K52
3820 IF I=5 THEN DLDYM1P=K11:DLDYM2P=K12:DDDYM1P=K21:DDDYM
     2P=K22:DYDYM1P=K31
3830 IF I=5 THEN DYDYM2P=K32:DPMDYM1P=K41:DPMDYM2P=K42:DRM
     DYM1P=K51:DRMDYM2P=K52
3840 IF I=6 THEN DLDRM1P=K11:DLDRM2P=K12:DDDRM1P=K21:DDDRM
     2P=K22:DYDRM1P=K31
3850 IF I=6 THEN DYDRM2P=K32:DPMDRM1P=K41:DPMDRM2P=K42:DYM
     DRM1P=K51:DYMDRM2P=K52
3860 IF I=7 THEN DDDL1N=K11:DDDL2N=K12:DYDL1N=K21:DYDL2N=K
     22:DPMDL1N=K31
3870 IF I=7 THEN DPMDL2N=K32:DYMDL1N=K41:DYMDL2N=K42:DRMDL
     1N=K51:DRMDL2N=K52
3880 IF I=8 THEN DLDD1N=K11:DLDD2N=K12:DYDD1N=K21:DYDD2N=K
     22:DPMDD1N=K31
3890 IF I=8 THEN DPMDD2N=K32:DYMDD1N=K41:DYMDD2N=K42:DRMDD
     1N=K51:DRMDD2N=K52
3900 IF I=9 THEN DLDY1N=K11:DLDY2N=K12:DDDY1N=K21:DDDY2N=K
     22:DPMDY1N=K31
3910 IF I=9 THEN DPMDY2N=K32:DYMDY1N=K41:DYMDY2N=K42:DRMDY
```

Figure A.16 BALCAL.BAS (cont.)

```
     1N=K51:DRMDY2N=K52
3920 IF I=10 THEN DLDPM1N=K11:DLDPM2N=K12:DDDPM1N=K21:DDDP
     M2N=K22:DYDPM1N=K31
3930 IF I=10 THEN DYDPM2N=K32:DYMDPM1N=K41:DYMDPM2N=K42:DR
     MDPM1N=K51
3940 IF I=10 THEN DRMDPM2N=K52
3950 IF I=11 THEN DLDYM1N=K11:DLDYM2N=K12:DDDYM1N=K21:DDDY
     M2N=K22:DYDYM1N=K31
3960 IF I=11 THEN DYDYM2N=K32:DPMDYM1N=K41:DPMDYM2N=K42:DR
     MDYM1N=K51
3970 IF I=11 THEN DRMDYM2N=K52
3980 IF I=12 THEN DLDRM1N=K11:DLDRM2N=K12:DDDRM1N=K21:DDDR
     M2N=K22:DYDRM1N=K31
3990 IF I=12 THEN DYDRM2N=K32:DPMDRM1N=K41:DPMDRM2N=K42:DY
     MDRM1N=K51
4000 IF I=12 THEN DYMDRM2N=K52
4010 RETURN   'GOBACK TO SAVE CALIBRATION DATA
```

Figure A.16 BALCAL.BAS (cont.)

APPENDIX B

TABLES

INITIAL SET-UP FOR THE MODEL 8255 TRANSDUCER AMPLIFIER

Amp. #	Component	Gain Set.	Filter Set.	Ex. Volt.
1	DRAG	MAX VAR.	1	+5.0
2	LIFT	MAX VAR.	1	+6.5
3	PITCH M.	1K	1	+5.0
4	YAW M.	MAX VAR.	1	+5.0
5	ROLL M.	MAX VAR.	1	+5.0
6	YAW	MAX VAR.	1	+5.0
7	AOA	1K	1	+5.0

Table B.1 Amplifier Set-Up

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
-0.071	0.113	-0.007	-0.011	0.004	0.186	0.000
0.280	-1.387	0.045	0.378	0.287	14.516	1.000
1.585	-2.601	0.049	0.891	0.625	29.482	2.000
1.664	-4.074	0.105	1.435	0.912	43.893	3.000
2.848	-5.052	0.127	1.874	1.278	59.583	4.000
4.271	-6.129	0.153	2.713	1.607	73.570	5.000
5.375	-7.449	0.242	3.158	1.854	88.797	6.000
6.020	-8.869	0.158	3.596	2.217	104.396	7.000
7.221	-10.076	0.192	4.368	2.589	119.488	8.000
7.526	-11.329	0.248	4.782	2.963	135.242	9.000
8.937	-12.496	0.242	5.261	3.110	152.621	10.000
9.933	-14.106	0.351	5.818	3.441	167.287	11.000
10.405	-15.140	0.399	6.286	3.812	183.977	12.000
12.029	-16.199	0.316	6.754	4.125	200.527	13.000
12.926	-17.125	0.479	7.583	4.458	216.548	14.000
12.718	-18.477	0.494	8.123	4.798	233.051	15.000
15.056	-19.601	0.468	8.699	5.111	250.094	16.000
16.137	-21.004	0.544	9.320	5.411	266.391	17.000
17.027	-22.130	0.520	9.872	5.768	283.582	18.000
18.145	-23.641	0.669	10.391	6.097	300.855	19.000
19.308	-24.989	0.553	11.033	6.387	316.969	20.000
18.069	-23.486	0.586	10.611	6.064	301.018	19.000
17.189	-22.140	0.536	9.914	5.795	284.159	18.000
16.028	-20.774	0.476	9.537	5.440	266.569	17.000
15.428	-19.346	0.429	8.877	5.157	250.179	16.000
13.816	-18.680	0.487	8.168	4.785	233.317	15.000
13.337	-17.414	0.519	7.745	4.570	216.317	14.000
12.021	-16.133	0.396	6.991	4.098	200.854	13.000
11.244	-15.067	0.369	6.782	3.864	184.612	12.000

Table B.2 Loading for Positive Lift

Table B.2 (cont.)

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
10.546	-13.455	0.356	5.974	3.451	167.852	11.000
8.900	-12.083	0.312	5.521	3.111	152.598	10.000
8.123	-10.766	0.337	5.090	2.977	135.368	9.000
7.659	-9.671	0.215	4.526	2.539	119.388	8.000
6.445	-8.578	0.218	3.930	2.258	104.478	7.000
5.031	-7.069	0.201	3.468	1.940	89.744	6.000
4.466	-5.886	0.155	3.104	1.681	73.372	5.000
3.732	-4.496	0.153	2.234	1.331	60.023	4.000
2.653	-3.641	0.095	1.762	1.008	44.615	3.000
1.772	-2.358	0.083	1.434	0.664	29.709	2.000
0.562	-1.266	0.044	0.676	0.299	14.989	1.000
-0.071	0.113	-0.007	-0.011	0.004	0.186	0.000

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
-0.320	0.100	-0.017	0.045	-0.035	0.706	0.000
-1.327	1.268	-0.048	-0.971	-0.378	-16.872	-1.000
-1.621	3.330	-0.067	-1.733	-0.870	-34.542	-2.000
-2.710	4.523	-0.084	-2.269	-1.220	-51.716	-3.000
-3.527	6.070	-0.101	-3.274	-1.508	-68.610	-4.000
-4.738	7.794	-0.123	-3.799	-1.905	-85.823	-5.000
-5.426	9.236	-0.138	-4.476	-2.282	-103.785	-6.000
-5.962	10.779	-0.148	-5.340	-2.705	-121.931	-7.000
-6.863	12.783	-0.171	-6.035	-3.054	-139.943	-8.000
-7.750	13.969	-0.189	-6.650	-3.531	-158.430	-9.000
-9.176	14.954	-0.202	-7.712	-3.891	-176.471	-10.000
-10.108	17.002	-0.219	-8.475	-4.268	-193.887	-11.000
-10.618	18.142	-0.239	-9.073	-4.682	-212.924	-12.000
-11.172	20.122	-0.260	-9.915	-5.074	-232.423	-13.000
-12.466	21.659	-0.268	-10.695	-5.559	-250.647	-14.000
-14.763	22.990	-0.273	-11.271	-5.905	-268.582	-15.000
-15.194	24.511	-0.302	-12.181	-6.299	-287.888	-16.000
-15.222	26.181	-0.311	-12.977	-6.741	-306.602	-17.000
-17.686	28.042	-0.342	-13.536	-7.032	-326.103	-18.000
-18.451	29.447	-0.347	-14.373	-7.589	-345.674	-19.000
-18.887	30.984	-0.362	-15.017	-8.085	-364.848	-20.000
-18.320	29.603	-0.343	-14.438	-7.639	-346.843	-19.000
-17.685	28.028	-0.324	-13.515	-7.074	-327.285	-18.000
-16.401	26.287	-0.315	-13.016	-6.726	-308.430	-17.000
-14.732	24.678	-0.291	-12.307	-6.352	-288.934	-16.000
-14.621	23.082	-0.275	-11.367	-5.954	-269.874	-15.000
-13.548	21.942	-0.260	-10.647	-5.516	-250.898	-14.000
-11.538	20.193	-0.238	-9.848	-5.233	-231.718	-13.000
-11.140	18.221	-0.231	-9.298	-4.807	-213.766	-12.000

Table B.3 Loading for Negative Lift

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
-9.829	17.183	-0.220	-8.528	-4.345	-195.012	-11.000
-8.389	15.609	-0.189	-7.825	-3.871	-177.525	-10.000
-8.427	13.805	-0.174	-6.761	-3.560	-158.393	-9.000
-7.429	12.775	-0.162	-6.053	-3.159	-140.009	-8.000
-6.237	10.903	-0.149	-5.317	-2.765	-122.401	-7.000
-5.430	9.379	-0.121	-4.467	-2.350	-104.339	-6.000
-4.382	7.697	-0.111	-3.946	-2.049	-86.772	-5.000
-3.342	6.110	-0.082	-3.249	-1.605	-68.951	-4.000
-3.271	4.588	-0.066	-2.276	-1.175	-52.168	-3.000
-2.342	3.567	-0.055	-1.613	-0.789	-33.987	-2.000
-0.866	1.625	-0.045	-1.019	-0.454	-16.524	-1.000
-0.320	0.100	-0.017	0.045	-0.035	0.706	0.000

Table B.3 (cont.)

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
0.144	0.064	-0.016	0.038	0.000	-0.213	0.000
0.103	1.435	-0.041	0.027	-0.269	0.227	1.000
0.063	2.570	-0.054	0.093	-0.322	0.109	2.000
0.088	3.640	-0.037	0.187	0.019	0.442	3.000
-0.409	4.079	-0.064	0.076	-0.263	0.363	4.000
-0.607	4.972	-0.066	-0.006	-0.318	0.644	5.000
-0.818	5.636	-0.080	0.013	-0.466	1.058	6.000
-1.108	6.678	-0.089	0.003	-0.640	1.445	7.000
-1.813	9.358	-0.154	-0.270	-1.701	1.871	8.000
-1.866	9.969	-0.153	-0.504	-1.907	2.325	9.000
-2.733	10.906	-0.173	-0.326	-2.143	2.299	10.000
-4.186	11.435	-0.170	-0.430	-2.392	2.327	11.000
-4.579	12.786	-0.189	-0.625	-2.617	3.325	12.000
-4.202	13.594	-0.164	-0.792	-2.862	3.698	13.000
-4.849	14.810	-0.198	-1.071	-3.161	3.726	14.000
-4.640	15.565	-0.178	-0.995	-3.176	4.644	15.000
-4.720	16.652	-0.196	-1.206	-3.442	5.416	16.000
-5.129	17.627	-0.209	-1.310	-3.709	5.589	17.000
-5.186	18.299	-0.186	-1.285	-3.797	4.722	18.000
-5.685	19.090	-0.209	-1.344	-4.120	5.255	19.000
-6.112	20.784	-0.211	-1.392	-4.351	5.452	20.000
-5.701	19.611	-0.226	-1.299	-4.143	5.224	19.000
-5.305	18.604	-0.230	-1.330	-3.984	5.608	18.000
-5.020	18.090	-0.221	-1.304	-3.826	5.542	17.000
-4.171	17.036	-0.224	-1.143	-3.464	5.131	16.000
-4.431	15.455	-0.190	-1.055	-3.137	4.508	15.000
-4.536	14.847	-0.219	-0.928	-3.045	3.552	14.000
-4.689	14.038	-0.231	-0.874	-2.962	3.460	13.000
-4.359	12.790	-0.200	-0.651	-2.715	2.729	12.000

Table B.4 Loading for Positive Drag

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
-3.948	11.490	-0.184	-0.309	-2.289	2.717	11.000
-3.196	10.758	-0.178	-0.205	-2.100	2.598	10.000
-2.793	9.849	-0.170	-0.168	-1.835	2.613	9.000
-2.011	8.958	-0.156	-0.114	-1.686	1.842	8.000
-1.308	7.860	-0.139	-0.139	-1.442	0.818	7.000
-1.224	6.379	-0.119	-0.026	-1.155	0.992	6.000
-1.152	5.406	-0.120	0.018	-1.046	0.674	5.000
-1.047	4.629	-0.107	0.054	-0.826	-0.111	4.000
-0.603	3.881	-0.075	0.090	-0.573	-0.176	3.000
-0.330	2.591	-0.059	0.206	-0.300	-0.188	2.000
0.118	1.637	-0.048	0.073	-0.116	-0.347	1.000
0.144	0.064	-0.016	0.038	0.000	-0.213	0.000

Table B.4 (cont.)

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
-0.047	0.128	0.003	0.021	-0.055	0.105	0.000
-0.268	-0.546	0.007	-0.527	-0.209	1.577	-1.000
-0.376	-1.706	0.010	-0.856	-0.241	2.934	-2.000
-0.259	-2.447	0.016	-0.987	-0.256	4.103	-3.000
-0.334	-3.405	0.004	-1.084	-0.304	5.706	-4.000
0.059	-4.182	0.007	-1.097	-0.211	6.613	-5.000
0.028	-5.573	0.010	-1.143	-0.250	8.271	-6.000
0.271	-6.284	0.016	-1.234	-0.194	10.216	-7.000
0.252	-7.019	0.010	-1.563	-0.178	11.656	-8.000
0.292	-7.862	0.011	-1.847	-0.193	13.351	-9.000
0.428	-9.494	0.023	-1.803	0.156	14.560	-10.000
0.445	-10.430	0.020	-2.081	0.066	15.356	-11.000
0.700	-11.191	0.013	-2.252	0.031	17.417	-12.000
0.895	-12.043	0.022	-2.321	0.188	18.810	-13.000
1.103	-13.295	0.028	-2.413	0.316	20.268	-14.000
1.036	-14.414	0.022	-2.561	0.281	21.610	-15.000
1.507	-15.153	0.026	-2.805	0.372	22.911	-16.000
1.392	-16.192	0.019	-3.038	0.460	24.032	-17.000
2.057	-17.759	0.021	-3.084	0.658	25.901	-18.000
2.038	-18.552	0.034	-3.238	0.573	27.110	-19.000
1.611	-18.979	0.030	-3.363	0.356	28.736	-20.000
1.627	-18.488	0.027	-3.308	0.508	27.292	-19.000
1.692	-17.609	0.030	-3.106	0.518	25.689	-18.000
1.757	-16.764	0.023	-2.835	0.592	23.895	-17.000
1.882	-15.817	0.024	-2.509	0.631	22.722	-16.000
1.971	-15.061	0.023	-2.351	0.867	21.581	-15.000
1.531	-13.449	0.024	-2.477	0.308	20.197	-14.000
1.392	-12.477	0.022	-2.359	0.343	18.803	-13.000
1.613	-11.454	0.016	-2.163	0.416	17.453	-12.000

Table B.5 Loading For Negative Drag

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
1.175	-10.663	0.021	-1.892	0.291	15.265	-11.000
1.551	-9.766	0.011	-1.656	0.388	14.555	-10.000
0.874	-8.346	0.016	-1.697	-0.193	13.526	-9.000
1.070	-7.236	0.023	-1.297	-0.056	11.567	-8.000
1.074	-6.647	0.007	-1.156	0.005	10.181	-7.000
0.948	-5.541	0.011	-1.158	-0.295	8.384	-6.000
1.084	-4.263	0.004	-1.046	-0.018	7.275	-5.000
0.611	-3.582	0.001	-0.860	-0.185	6.052	-4.000
0.807	-2.663	0.009	-0.675	-0.180	4.407	-3.000
0.526	-1.796	0.017	-0.354	-0.274	3.091	-2.000
0.019	-0.502	0.003	-0.055	-0.192	1.860	-1.000
-0.047	0.128	0.003	0.021	-0.055	0.105	0.000

Table B.5 (cont.)

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
0.166	0.100	0.002	0.008	-0.008	0.120	0.000
-2.221	1.450	0.952	-1.108	-3.126	2.045	1.000
-4.584	2.770	1.958	-2.031	-6.398	4.872	2.000
-8.008	4.316	2.973	-3.360	-9.619	6.369	3.000
-10.188	5.473	3.961	-4.005	-12.859	8.063	4.000
-12.156	7.485	4.940	-4.961	-16.118	9.770	5.000
-14.632	9.098	5.946	-5.618	-19.031	11.992	6.000
-17.930	10.452	6.936	-6.462	-22.242	14.249	7.000
-21.630	12.019	7.931	-7.644	-25.453	16.619	8.000
-24.214	13.765	8.974	-8.448	-28.676	18.883	9.000
-27.802	15.002	9.941	-9.427	-31.652	19.868	10.000
-30.794	16.649	10.888	-10.068	-34.573	21.895	11.000
-34.323	18.084	11.901	-11.239	-37.945	24.728	12.000
-37.631	19.661	12.847	-12.372	-40.886	26.277	13.000
-41.456	21.630	13.835	-13.000	-43.932	28.876	14.000
-47.040	23.856	14.861	-14.416	-47.777	30.882	15.000
-50.053	25.205	15.802	-14.841	-50.103	34.085	16.000
-52.008	26.471	16.799	-15.510	-53.119	36.597	17.000
-58.790	28.820	17.783	-16.971	-56.886	38.280	18.000
-62.650	30.278	18.755	-17.679	-59.750	40.188	19.000
-65.594	31.940	19.738	-18.501	-62.337	43.755	20.000
-64.978	31.710	19.477	-18.377	-61.613	42.953	19.000
-59.463	28.995	17.908	-16.992	-57.049	39.051	18.000
-55.590	28.391	17.480	-16.207	-55.472	38.018	17.000
-50.338	24.991	15.904	-14.850	-50.364	34.373	16.000
-48.193	24.032	15.467	-14.518	-49.057	33.498	15.000
-41.215	21.817	13.916	-13.116	-44.146	29.675	14.000
-37.583	20.231	12.990	-12.591	-41.367	27.194	13.000
-36.297	18.948	12.194	-11.421	-39.036	25.729	12.000

Table B.6 Loading for Positive Yaw

Table B.6 (cont.)

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
-31.183	17.109	11.073	-10.167	-35.216	23.473	11.000
-27.492	15.307	10.141	-9.484	-31.984	21.584	10.000
-23.788	13.713	9.022	-8.394	-28.689	18.667	9.000
-21.227	11.942	8.038	-7.564	-25.422	16.641	8.000
-17.927	10.710	7.143	-6.711	-22.570	14.574	7.000
-14.556	9.361	6.044	-5.639	-19.039	12.346	6.000
-12.173	7.227	5.092	-4.848	-15.843	10.535	5.000
-9.954	5.821	3.997	-3.934	-12.944	8.368	4.000
-7.879	4.408	3.005	-3.093	-9.650	6.769	3.000
-4.531	3.227	2.051	-1.970	-6.483	4.986	2.000
-2.098	1.751	0.977	-1.027	-3.179	2.248	1.000
0.166	0.100	0.002	0.008	-0.008	0.120	0.000

Table B.7 Loading for Negative Yaw

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
-0.055	0.059	0.001	0.063	0.011	0.093	0.000
2.967	-1.059	-0.958	0.814	2.800	-1.335	-1.000
5.120	-3.017	-1.967	1.483	5.865	-3.260	-2.000
9.351	-3.996	-2.962	2.289	8.951	-5.569	-3.000
12.379	-5.175	-4.004	2.978	12.092	-6.474	-4.000
15.312	-6.789	-5.002	3.835	15.341	-7.712	-5.000
18.821	-8.044	-5.984	4.533	18.385	-8.733	-6.000
22.298	-10.201	-6.985	5.425	21.697	-10.471	-7.000
25.668	-11.360	-8.079	6.394	25.109	-12.095	-8.000
30.029	-12.708	-9.029	7.141	28.223	-13.451	-9.000
33.394	-14.695	-10.008	7.938	31.286	-14.795	-10.000
37.067	-15.702	-10.882	8.985	34.161	-15.909	-11.000
40.183	-17.035	-11.851	9.548	37.032	-17.473	-12.000
43.295	-18.697	-12.867	10.347	40.414	-18.254	-13.000
48.918	-19.892	-13.817	11.280	43.732	-19.292	-14.000
52.645	-21.816	-14.834	12.138	46.734	-21.012	-15.000
56.876	-23.024	-15.921	13.090	49.950	-22.206	-16.000
61.506	-25.086	-16.906	14.010	53.449	-23.491	-17.000
66.405	-26.855	-18.014	15.040	57.019	-25.279	-18.000
69.930	-28.047	-18.886	16.051	59.726	-25.959	-19.000
74.306	-30.189	-19.884	16.967	63.041	-26.666	-20.000
70.803	-28.908	-19.095	16.286	60.537	-26.031	-19.000
68.264	-27.218	-18.107	15.610	57.885	-24.959	-18.000
62.162	-25.538	-16.925	14.443	54.100	-23.205	-17.000
58.846	-24.198	-16.093	13.560	51.431	-22.340	-16.000
52.757	-22.058	-14.995	12.463	47.182	-21.090	-15.000
49.430	-20.316	-13.840	11.546	44.132	-19.248	-14.000
43.593	-18.641	-12.946	10.558	40.542	-18.102	-13.000
40.778	-17.547	-11.946	9.856	37.670	-16.751	-12.000

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS LBS.
37.661	-15.865	-11.075	9.213	34.940	-15.773	-11.000
34.570	-14.791	-10.072	8.389	31.824	-14.770	-10.000
29.287	-12.187	-8.955	7.083	27.878	-13.305	-9.000
26.086	-11.401	-8.010	6.463	25.056	-12.264	-8.000
21.526	-9.723	-7.075	5.484	21.682	-10.212	-7.000
18.626	-7.830	-6.014	4.813	18.554	-8.891	-6.000
15.819	-6.804	-5.069	4.066	15.600	-7.661	-5.000
12.420	-5.390	-4.040	3.138	12.376	-6.073	-4.000
9.107	-3.991	-3.046	2.599	9.184	-5.048	-3.000
5.517	-2.901	-2.037	1.799	6.175	-3.386	-2.000
2.913	-1.042	-0.996	0.887	3.049	-1.494	-1.000
-0.055	0.059	0.001	0.063	0.011	0.093	0.000

Table B.7 (cont.)

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS IN.-LBS.
0.214	0.211	-0.001	-0.025	-0.009	0.080	0.000
9.468	28.547	-0.001	13.012	8.673	-11.054	20.000
18.834	58.099	-0.033	26.355	17.266	-22.066	40.000
29.181	87.189	-0.058	40.271	25.624	-32.834	60.000
39.729	116.142	-0.082	54.581	34.027	-43.690	80.000
51.132	145.804	-0.109	69.702	42.167	-53.712	100.000
64.056	176.691	-0.123	85.694	50.777	-65.525	120.000
76.604	206.900	-0.128	101.923	58.724	-76.172	140.000
88.111	234.610	-0.122	117.496	66.495	-84.066	160.000
100.781	265.683	-0.186	134.947	74.174	-94.863	180.000
114.512	297.252	-0.181	153.081	82.029	-105.526	200.000
101.755	266.397	-0.137	135.461	74.735	-96.058	180.000
88.159	235.886	-0.133	117.651	66.528	-85.026	160.000
76.039	207.251	-0.126	101.846	58.726	-75.896	140.000
63.970	177.839	-0.114	85.756	50.817	-66.951	120.000
51.714	146.766	-0.090	69.630	42.436	-53.868	100.000
40.502	116.353	-0.058	54.721	34.242	-43.078	80.000
29.586	86.839	-0.053	39.953	25.635	-31.998	60.000
19.403	58.176	-0.039	26.159	17.333	-21.830	40.000
9.582	28.496	-0.017	12.768	8.688	-10.569	20.000
0.214	0.211	-0.001	-0.025	-0.009	0.080	0.000

Table B.8 Loading for Positive Pitching Moment

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS IN.-LBS.
-0.022	0.053	-0.013	-0.085	0.002	1.204	0.000
-9.342	-22.299	-0.034	-20.149	-9.753	8.902	-20.000
-20.459	-46.536	0.005	-40.398	-19.304	18.062	-40.000
-31.843	-70.894	-0.030	-59.988	-28.202	25.567	-60.000
-46.272	-97.552	-0.051	-80.013	-37.564	34.076	-80.000
-60.649	-124.862	-0.052	-99.914	-46.601	42.721	-100.000
-77.102	-153.984	-0.012	-119.811	-55.450	51.347	-120.000
-95.409	-184.075	-0.005	-140.690	-64.299	59.544	-140.000
-113.896	-213.973	-0.019	-160.273	-72.931	67.517	-160.000
-134.509	-245.938	-0.013	-180.202	-81.501	75.865	-180.000
-156.696	-279.220	-0.016	-200.807	-90.054	84.861	-200.000
-135.589	-246.365	0.044	-180.564	-81.905	76.076	-180.000
-110.857	-215.607	-0.052	-160.126	-72.143	67.292	-160.000
-95.290	-183.924	0.040	-140.576	-64.833	59.076	-140.000
-78.035	-154.095	-0.167	-120.342	-55.966	50.528	-120.000
-59.298	-126.386	-0.026	-99.877	-46.182	41.785	-100.000
-45.867	-97.905	-0.055	-80.014	-37.366	33.798	-80.000
-31.443	-71.330	-0.008	-59.783	-28.420	25.332	-60.000
-19.286	-46.787	-0.048	-40.361	-18.893	16.988	-40.000
-9.088	-22.485	-0.025	-20.020	-9.698	8.443	-20.000
-0.022	0.053	-0.013	-0.085	0.002	1.204	0.000

Table B.9 Loading for Negative Pitching Moment

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS IN.-LBS.
0.066	-0.131	-0.008	0.025	-0.110	0.212	0.000
5.562	-4.377	0.211	0.835	4.240	1.341	4.500
11.857	-9.049	0.370	2.083	8.887	2.601	9.000
18.013	-13.622	0.495	3.150	13.489	3.331	13.500
24.747	-18.318	0.626	3.910	18.382	4.961	18.000
31.940	-22.505	0.829	5.555	22.643	4.846	22.500
37.782	-26.961	1.037	6.809	27.220	6.147	27.000
44.619	-32.413	1.032	7.773	32.072	6.461	31.500
53.843	-36.639	1.193	9.141	36.627	7.297	36.000
59.427	-41.664	1.599	9.996	40.472	8.143	40.500
69.069	-46.790	2.142	11.322	45.119	9.719	45.000
60.551	-41.774	1.587	10.158	40.805	8.513	40.500
53.264	-36.577	1.493	9.025	36.165	7.753	36.000
45.255	-32.552	1.278	7.913	32.021	7.069	31.500
37.751	-27.066	1.106	6.789	27.180	5.832	27.000
31.770	-22.787	0.996	5.544	22.547	5.388	22.500
24.763	-18.305	0.602	4.545	18.441	5.226	18.000
17.238	-13.475	0.505	3.055	13.512	4.338	13.500
12.701	-8.738	0.338	2.273	9.097	2.429	9.000
5.932	-4.225	0.163	1.185	4.491	1.155	4.500
0.066	-0.131	-0.008	0.025	-0.110	0.212	0.000

Table B.10 Loading for Positive Yawing Moment

Table B.11 Loading for Negative Yawing Moment

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS IN.-LBS.
-0.364	-0.032	-0.028	0.028	-0.027	0.781	0.000
-4.753	4.450	-0.176	-1.317	-4.459	1.095	-4.500
-9.841	9.326	-0.366	-2.934	-8.975	1.589	-9.000
-14.898	13.985	-0.540	-4.137	-13.347	2.262	-13.500
-20.846	18.948	-0.759	-5.256	-18.131	2.682	-18.000
-27.429	23.529	-0.905	-6.851	-22.375	3.800	-22.500
-33.057	28.182	-0.931	-8.447	-26.892	4.200	-27.000
-41.042	33.270	-0.940	-9.815	-31.754	4.779	-31.500
-48.163	38.046	-1.406	-11.161	-35.695	4.987	-36.000
-56.084	42.681	-1.179	-12.434	-40.492	6.500	-40.500
-63.355	47.777	-1.296	-13.990	-44.956	6.613	-45.000
-55.843	42.721	-1.134	-12.621	-40.644	6.260	-40.500
-48.025	38.026	-1.340	-11.161	-35.826	5.420	-36.000
-40.568	33.183	-1.021	-9.776	-31.786	4.654	-31.500
-32.476	28.313	-0.984	-8.537	-26.865	4.089	-27.000
-26.527	23.464	-0.671	-6.919	-22.446	3.769	-22.500
-20.665	18.716	-0.714	-5.330	-18.099	3.247	-18.000
-15.310	14.012	-0.561	-4.251	-13.569	2.482	-13.500
-9.742	9.281	-0.337	-2.996	-9.067	2.204	-9.000
-4.863	4.281	-0.192	-1.549	-4.485	1.514	-4.500
-0.364	-0.032	-0.028	0.028	-0.027	0.781	0.000

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS IN.-LBS.
-0.261	0.082	-0.012	-0.141	-0.062	0.683	0.000
2.400	-0.977	-0.017	-1.133	-0.787	12.072	11.500
5.066	-2.106	-0.098	-2.068	-1.360	23.143	23.000
7.477	-3.170	-0.113	-3.123	-2.272	34.326	34.500
10.411	-4.364	-0.166	-4.087	-2.780	47.042	46.000
12.944	-5.136	-0.225	-4.966	-3.525	57.242	57.500
16.316	-6.254	-0.233	-5.749	-4.195	68.583	69.000
19.773	-7.130	-0.317	-6.933	-4.964	80.292	80.500
22.995	-8.211	-0.303	-7.909	-5.483	93.388	92.000
26.416	-9.415	-0.308	-9.060	-6.342	104.096	103.500
29.380	-10.585	-0.388	-9.725	-6.723	115.821	115.000
26.436	-9.063	-0.308	-9.225	-6.460	102.672	103.500
22.941	-7.841	-0.417	-7.918	-5.679	91.494	92.000
20.190	-6.985	-0.274	-7.125	-5.156	78.468	80.500
16.915	-6.002	-0.260	-5.547	-4.083	68.825	69.000
14.206	-5.466	-0.241	-5.042	-3.244	56.446	57.500
10.928	-3.948	-0.176	-4.108	-2.717	46.912	46.000
8.552	-3.227	-0.136	-3.055	-2.002	35.616	34.500
4.663	-2.367	-0.098	-2.280	-1.340	23.271	23.000
2.454	-1.191	-0.033	-1.165	-0.711	12.422	11.500
-0.261	0.082	-0.012	-0.141	-0.062	0.683	0.000

Table B.12 Loading for Positive Rolling Moment

L LBS.	D LBS.	Y LBS.	PM IN.-LBS.	YM IN.-LBS.	RM IN.-LBS.	LOADS IN.-LBS.
-0.188	0.053	-0.005	-0.023	-0.024	0.210	0.000
-2.414	0.822	0.022	0.818	0.688	-10.239	-11.500
-5.413	1.281	0.050	1.275	1.289	-22.723	-23.000
-7.829	2.468	0.068	2.224	1.917	-34.816	-34.500
-10.429	3.520	0.084	2.837	2.760	-45.299	-46.000
-14.950	3.991	0.245	3.540	3.525	-57.070	-57.500
-17.179	4.586	0.102	4.271	4.225	-68.102	-69.000
-20.313	5.433	0.136	5.085	4.988	-79.611	-80.500
-23.743	6.551	0.161	5.967	5.424	-91.562	-92.000
-26.884	7.181	0.195	6.562	6.349	-102.372	-103.500
-31.111	8.611	0.286	7.206	6.775	-113.554	-115.000
-27.668	7.449	0.315	6.546	6.124	-105.360	-103.500
-23.715	6.705	0.242	6.022	5.623	-94.574	-92.000
-20.660	5.896	0.214	5.060	4.659	-82.659	-80.500
-18.021	5.410	0.064	4.261	4.069	-71.081	-69.000
-14.706	4.519	0.125	3.446	3.421	-60.337	-57.500
-10.793	3.840	0.120	2.945	2.816	-46.204	-46.000
-8.334	2.809	0.094	2.160	2.114	-34.994	-34.500
-5.582	1.321	0.016	1.432	1.450	-22.390	-23.000
-3.077	0.825	0.015	0.800	0.792	-12.264	-11.500
-0.188	0.053	-0.005	-0.023	-0.024	0.210	0.000

Table B.13 Loading for Negative Rolling Moment

LIST OF REFERENCES

1. Mair, R. Scott, Wind Tunnel Drag Evaluations of Helicopter Nose Sections, Naval Postgraduate School Thesis, Monterey, California, 1985.

2. Sargent, Christopher L., Wind Tunnel Drag Evaluation of Helicopter Tail Sections, Naval Postgraduate School Thesis, Monterey, California, 1985.

3. Giordano, Frank R. and Weir, Maurice D., A First Course in Mathematical Modeling, p. 114, Brooks & Cole, 1985

BIBLIOGRAPHY

Department of Aeronautics, Laboratory Manual for Low-Speed Wind-Tnuuel Testing, Naval Postgraduate School, Monterey, California, October, 1983.

Gorlin, S.M. and Slezinger, I.I., Wind Tunnels and Their Instrumentation, Israel Program for Scientific Translations, 1966.

Park, Chan S., Interactive Micrcomputer Graphics, Addison Wesley, 1985.

Saunders, George H., Dynamics of Helicopter Flight, Wiley Interscience, 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia, 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5000	1
4.	Prof. Donald M. Layton, Code 67-LN Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5000	2
5.	Ted Dunton, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5000	1
6.	Lt. Patrick A. Witt 2008 Inverrary Ct. Virginia Beach, Virginia 23456	2

21684

Thesis
W6682 Witt
c.1 Development of a data acquisition system to aid in the aerodynamic study of various helicopter configurations.
31 OCT 89 36547
31 OCT 89 36547

21684

Thesis
W6682 Witt
c.1 Development of a data acquisition system to aid in the aerodynamic study of various helicopter configurations.

DUDLEY KNOX LIBRARY
3 2768 00012975 3